

Tight Network Topology Dependent Bounds on Rounds of Communication^{*}

ARKADEV CHATTOPADHYAY[§] MICHAEL LANGBERG[†] SHI LI[‡] ATRI RUDRA[‡]

[§]School of Technology and Computer Science,
Tata Institute of Fundamental Research
arkadev.c@tifr.res.in

[†]Department of Electrical Engineering,
University at Buffalo, SUNY
mikel@buffalo.edu

[‡]Department of Computer Science and Engineering,
University at Buffalo, SUNY
{shil, atri}@buffalo.edu

Abstract

We prove tight network topology dependent bounds on the round complexity of computing well studied k -party functions such as set disjointness and element distinctness. Unlike the usual case in the CONGEST model in distributed computing, we fix the function and then vary the underlying network topology. This complements the recent such results on total communication that have received some attention. We also present some applications to distributed graph computation problems.

Our main contribution is a proof technique that allows us to reduce the problem on a general graph topology to a relevant two-party communication complexity problem. However, unlike many previous works that also used the same high level strategy, we do *not* reason about a two-party communication problem that is induced by a cut in the graph. To ‘stitch’ back the various lower bounds from the two party communication problems, we use the notion of timed graph that has seen prior use in network coding. Our reductions use some tools from Steiner tree packing and multi-commodity flow problems that have a delay constraint.

^{*}AC’s research is partially supported by a Ramanujan fellowship of the DST. ML’s research is supported in part by NSF-CCF1526771. SL’s research is partly supported by NSF grant CCF-1566356. AR’s research is supported in part by NSF grant CCF-1319402.

1 Introduction

In this paper, we prove bounds on the number of rounds needed to compute a given function in a distributed manner. In our paper a problem is a tuple (f, G, K) , where $G = (V, E)$ is the underlying communication graph (which is assumed to be *undirected*), $K \subseteq V$ is a set of $k \stackrel{\text{def}}{=} |K|$ terminals (or players), and we are interested in computing the function $f : (\{0, 1\}^n)^K \rightarrow \{0, 1\}$: i.e. all terminals in K need to know the final answer after the protocol is done.¹ Unless stated otherwise, the k inputs are assigned in worst-case manner to the terminals in K .

All communication in a protocol is point-to-point (as opposed to the broadcast mode of communication) and a bit transmitted over an edge $e = (u, v)$ is private to u and v . Further, we assume a synchronous model and in each *round*, each node $u \in V$ sends a (potentially different) bit² to each of its neighbors. We assume the two directions of an edge (u, v) can be used simultaneously. We will further assume that the protocols have full knowledge of G and all nodes (for randomized protocols) use public randomness.³ In this paper, we are interested in the round complexity: i.e. the total number of rounds needed by a protocol to compute the output. Note that this notion corresponds to the time taken by the distributed protocol to compute the answer. Given a problem \mathcal{P} we will use $R_\epsilon(\mathcal{P})$ to denote the minimum number of rounds needed for the worst-case input of any randomized protocol that errs on all input with probability at most ϵ . WLOG for randomized protocols one can assume that $\epsilon = 1/3$ and we will in most cases refer to $R_{1/3}(\mathcal{P})$ by just $R(\mathcal{P})$. Note that $R_0(\mathcal{P})$ denotes the *deterministic* round complexity.⁴ To simplify our presentation we will ignore in our bounds poly-logarithmic factors in both the size of G and n . In particular, we will use the notation $\tilde{O}(\cdot)$, $\tilde{\Omega}(\cdot)$ and $\tilde{\Theta}(\cdot)$ to denote the usual asymptotic notation that ignore poly-log factors (in size of G and n).

Our model above is very similar to the well studied CONGEST model in distributed computing [Pel00] with the following differences. First, for proving upper bounds on the CONGEST model, it is assumed that a node in V only knows about its neighbors while in our setup we assume that the protocol knows the structure of G . This makes our lower bounds potentially stronger (though this makes our upper bounds weaker than the distributed protocol bounds in the CONGEST model). Second, typically in the distributed computing literature the function f itself depends on the underlying network G (e.g. check if a given subgraph of G is a spanning tree [DSHK⁺12]) while in our setup the function f is independent of the network topology G . This assumption makes sense in the current state of affairs where many such functions are computed in a distributed manner over the same network. Recent works (including those of Drucker et al. [DKO14] and Klauck et al. [KNPR15]) have proved bounds for functions for the special case where G is the complete graph. Finally, in most of the existing work it is assumed that $K = V$, while we consider the more general case when $K \subseteq V$. This more general case makes sense e.g. in a data warehouse where any given function that needs to be computed could only depend on inputs that are stored at some subset of the servers.

¹It turns out if one terminal knows the answer then it can send the answer to all others via a simple Steiner tree based protocol whose cost is dominated by all our bounds.

²By creating parallel edges, our results extend to the case where in each round each edge $e \in E$ can send c_e bits. However, for notational simplicity we will only consider the case of $c_e = 1$ in this paper; that is, G is a simple graph.

³Since all parties know the entire topology G , then one can generalize Newman's argument [New91] to our setting. In our case, in the private randomness protocol corresponding to the original public randomness protocol, one party will need to send $O(\log nk)$ bits to all other $k - 1$ parties. This can be accomplished by a simple Steiner tree protocol, whose cost can be absorbed in all of our bounds. We will stick with public randomness since it makes the description of our protocols easier to follow.

⁴We note that in communication complexity literature, R_0 is used to denote the zero-error randomized communication complexity but we use this convention since it makes our theorem statements cleaner.

Recently there has been work that deals with the graph communication model as above but instead of minimizing the round complexity, these results are for the case of minimizing the *total communication* of the protocols. (We note that the total communication corresponds to the *message complexity* of distributed protocols.) Most of the work in this area has been for specific classes of G . For example, the early work of Tiwari [Tiw87] considered deterministic total communication complexity on cases of G being a path, grid or ring graph. There has been a recent surge of interest for proving lower bounds on total communication for the case when G is a star [PVZ12, WZ12, WZ13, BEO⁺13, WZ14, CM15]. This work was generalized to arbitrary topology by Chattopadhyay et al. [CRR14] who proved tight bounds for certain functions for *all* network topologies. A followup work extended the results to some more functions [CR15].

Both of these strands of work (on round complexity and total communication) coincide for the special case when G is just an edge. Note that in this case we have two players and the model coincides with the very well studied model of two-party communication complexity introduced by Yao [Yao79], which has proved to be an extremely worthwhile model to study with applications in diverse areas of theoretical computer science.

Given the importance of round complexity in distributed computing, it is natural to ask

Can we prove tight topology sensitive bounds for round complexity?

We would like to point out that optimal protocols for total communication need not be optimal for round complexity and vice-versa. To see this, consider the case where G contains two terminals $\{a, b\}$ and many parallel edge-disjoint paths between a and b : there is one path of length 1, and \sqrt{n} paths of length \sqrt{n} . a receives n bits and wants to send those n bits to b . The optimal protocol for total communication would be to send the n bits on the length-1 path, which has $O(n)$ total communication but takes $\Omega(n)$ rounds. On the other hand, an (almost) optimal protocol in terms of number of rounds would be splitting the n bits into \sqrt{n} blocks of \sqrt{n} bits and send each block using one of the \sqrt{n} paths of length \sqrt{n} . This protocol has round complexity $O(\sqrt{n})$ but total communication $\Omega(n\sqrt{n})$.

In this work, we prove tight bounds on round complexity for several families of functions. We believe that the proof techniques presented, and not just the concrete results, are of independent interest. Pretty much all of the previous work in the total communication regime proved their lower bounds via two steps. The first step was to ‘divide’ up the problem into a bunch of two party communication complexity problems. The second step is to ‘stitch’ together the lower bounds for these two party communication problems. Our proofs also have the same two step structure but our implementations of both these steps are very different. The first step in previous works is implemented by constructing a family of cuts and then considering the two-party problem induced on each cut. In our proofs, we consider a more general set of edges E' (which might not form a cut) and then simulate our original protocol on G projected down to E' via a two-party communication protocol. The second step in previous work used a common hard distribution across all chosen cuts and then used linearity of expectation to ‘add up’ the lower bounds. In contrast, we use the notion of a *timed graph* (that is independent of the hard distributions) so that we can use different hard distributions for the different two party communication problems to deduce something about the *same* timed graph. The stitching then occurs by proving various ‘gluing’ results on Steiner tree packing and multi-commodity flow problems on graphs. This difference allows us to prove lower bounds for both randomized and deterministic protocols with the same proof while e.g. the results of [CRR14] could not prove tight deterministic bounds for functions whose zero-error randomized complexity is much smaller than its deterministic complexity.

1.1 Overview of our results

We prove our bounds for two classes of functions, as in [CR15]. Roughly speaking in the total communication setting, one class has an optimal protocol that combines inputs upwards on a Steiner tree and in the second class of problems the optimal protocol involves all players sending their inputs to a designated node. Next, we define two functions that are representatives of these two classes. (See Theorems 14 and 19 for the exact definitions of these two classes.)

We start our overview with the well studied k -party set disjointness problem, which is defined as follows. Each player $u \in K$ gets a string $\mathbf{x}_u \in \{0, 1\}^n$ (which can be thought of as a subset of $[n] \stackrel{\text{def}}{=} \{1, 2, \dots, n\}$) and the output is

$$\text{DISJ}_{K,n}(\{\mathbf{x}_u\}_{u \in K}) = \bigvee_{i \in [n]} \bigwedge_{u \in K} \mathbf{x}_u[i],$$

i.e. the output is 1 if and only if all the k sets have an element in common.

For $\text{DISJ}_{K,n}$ as shown in [CR15], the optimal protocol (up to poly-logarithmic factors) for total communication is to first compute the minimum Steiner tree on G with K as the set of terminals and then to compute the intersection of the k sets in a bottom-up fashion. For the round complexity, it seems natural to try this scheme in ‘parallel’: i.e. try to *pack* as many edge disjoint Steiner trees of small diameter as possible and to compute the set intersection on appropriate parts of the universe $[n]$ up the trees in parallel. It turns out that this is indeed the optimal protocol for round complexity. We prove the following result (where $\text{ST}(G, K, \Delta)$ denotes the optimal value of Steiner tree packing with terminals K and diameter Δ in G ; formal definition appears in Section 2):

Theorem 1. *For any graph G and subset of players K , we have for every $\epsilon \geq 0$*

$$R_\epsilon(\text{DISJ}_{K,n}, G, K) = \tilde{\Theta}\left(\min_{\Delta \in [|V|]} \left(\frac{n}{\text{ST}(G, K, \Delta)} + \Delta\right)\right).$$

The other function is the element distinctness problem (shortened to ED), which is defined as follows. Each player $u \in K$ gets a string $\mathbf{x}_u \in \{0, 1\}^n$ (which can be thought as a number in $[0, 2^n - 1]$) and the output is

$$\text{ED}_{K,n}(\{\mathbf{x}_u\}_{u \in K}) = \bigwedge_{u \neq v \in K} \mathbf{x}_u \neq \mathbf{x}_v.$$

For $\text{ED}_{K,n}$ as shown in [CRR14], the optimal randomized protocol for total communication is for the k players to send the hash of their inputs to the median node w.r.t. K in G . A natural protocol would be to run a multi-commodity flow problem where the demands correspond to each of the k players sending their bits to the median node. However, it turns out that this is *not* optimal for round complexity. Intuitively the main reason this fails is because the median node has too much incoming flow. The next natural idea would be to somehow have a different multi-commodity flow problem where each node has a ‘balanced load’. Indeed we are able to show this to be possible by using a small circuit for $\text{ED}_{K,n}$ as our guide. Let $\tau_{\text{MCF}}(G, K, n')$ denote the smallest number of rounds τ needed to simultaneously route n'/k units flow from u to v for every $u, v \in K$. Then we show that

Theorem 2. *For any G and K , we have for any constant $\epsilon > 0$*

$$R_\epsilon(\text{ED}_{K,n}, G, K) = \tilde{\Theta}(\tau_{\text{MCF}}(G, K, 1))$$

and

$$R_0(\text{ED}_{K,n}, G, K) = \tilde{\Theta}(\tau_{\text{MCF}}(G, K, n)).$$

In particular, we generalize the construction in Drucker et al. [DKO14] to show how to convert any bounded fan-in and fan-out circuit for any function f into a protocol for f . Drucker et al. proved such a result for the special case of G being the complete graph.⁵ More specifically, we show that

Lemma 3. *Let $f : (\{0, 1\}^n)^k \rightarrow \{0, 1\}$ have a circuit with constant fan-in and constant fan-out gates and depth d . Further, each level $i \in [d]$ has s_i gates in it (and let $s = \sum_{i=1}^d s_i$). Then*

$$R_0(f, G, K) \leq \sum_{i=1}^d \tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{s_i}{k}\right)\right).$$

Finally, we can upper bound the above by $\tilde{O}(d \cdot \tau_{\text{MCF}}(G, K, \frac{s}{k}))$ as well as $\tilde{O}(\frac{s}{k} \cdot \tau_{\text{MCF}}(G, K, 1))$.

Like the results of Drucker et al., this connection implies a barrier to proving quantitatively better lower bounds. In particular, if we could exhibit an explicit function f that we could prove requires asymptotically larger number of rounds than $\tilde{O}(d \cdot \tau_{\text{MCF}}(G, K, n))$, then we would have shown a super-linear size lower bound for circuits computing f with depth d . This extends the results in [DKO14], which showed that for the clique topology proving any super-constant lower bound on rounds for an explicit function f will imply a corresponding (new) circuit lower bounds.

We also apply our general lower bounds to prove lower bounds for the following distributed graph problems. In these problems each player $u \in K$ gets a graph H_u as input and the goal is to check if the overall graph $H = \bigcup_{u \in K} H_u$ has certain properties. In particular, we consider the following four problems that check if H (i) is connected, (ii) contains a triangle, (iii) is acyclic, (iv) is connected. We show a lower bound of $\tilde{\Omega}\left(\tau_{\text{MCF}}\left(G, K, \frac{|V(H)| + |E(H)|}{k}\right)\right)$. Our lower bounds extend some of the lower bounds in [KNPR15] to general topologies. In particular, we generalize the lower bounds for connectivity to general topologies while [KNPR15] does not provide any lower bounds for the triangle detection problem. (However, we note that [KNPR15]’s lower bound for connectivity holds for random distribution of H while our lower bounds assume worst-case distribution. So our lower bounds are proven in a weaker setting.) We also show, by simple adaptation of upper bounds in [DKO14, KNPR15], that as long as H is large, these bounds are also tight.

Finally, we highlight a technical result that we believe is of independent interest. We will use $R_\epsilon^{(2)}(f)$ to denote the randomized round complexity for the two party case (we assume Alice and Bob can send a bit to each other simultaneously in each round), where we allow Alice and Bob to have inputs of different sizes. We will also need to consider $R_\epsilon^{(-)}(f)$ for the *one-way* round complexity where (say) Alice (or Bob) sends a single message to Bob (Alice resp.) and Bob (Alice resp.) computes the answer based solely on the single message he (she resp.) received from Alice (Bob resp.) as well as his (her resp.) input. Let $\tau_{\text{route}}(G, \{u, v\}, n')$ denote the minimum number of rounds in which u can route n' bits to v in G ; since G is undirected, this is the same as the minimum number of rounds in which v can route n' bits to u in G ; thus the notation is well-defined.

Theorem 4. *For any function $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, and any graph G we have that*

$$\tau_{\text{route}}(G, \{a, b\}, R_\epsilon^{(2)}(f)) \leq 4R_\epsilon(f, G, \{a, b\}).$$

Notice the above inequality implies that $\frac{\tau_{\text{route}}(G, \{a, b\}, R_\epsilon^{(-)}(f))}{R_\epsilon(f, G, \{a, b\})} \leq 4 \left\lceil \frac{R_\epsilon^{(-)}(f)}{R_\epsilon^{(2)}(f)} \right\rceil$, by Claim 8 (stated in Section 2). The technical result implies that when using the obvious one-way communication algorithm to solve f for the case of $k = 2$, the penalty we incur on *any* graph is no worse than a constant factor when used on the case when G is just the edge (u, w) (i.e. the traditional two party communication complexity setting).

⁵However, [DKO14] do not lose any $\tilde{O}(1)$ factors like we do.

1.2 Overview of our proof techniques.

We now present an overview of our proof techniques specialized to the case of $\text{DISJ}_{K,n}$ and $\text{ED}_{K,n}$. To begin with we will assume that n is much larger than the size of G . In this case the common way to prove a lower bound is via the so called *communication bottleneck* argument: if m bits have to be transmitted over a cut C with $\delta(C)$ crossing edges, then the obvious lower bound on round complexity is $\frac{m}{|\delta(C)|}$.

We begin with $\text{DISJ}_{K,n}$. Using the communication bottleneck argument and the linear lower bound on the two-party communication complexity of DISJ [Raz92], we get a lower bound of $\Omega\left(\frac{n}{\lambda_K(G)}\right)$ for $\text{DISJ}_{K,n}$ (where $\lambda_K(G)$ is the size of the min cut separating K). For the upper bound, we invoke a result of Lau [Lau07] to argue that we can pack $T = \Omega(\lambda_K(G))$ many *edge disjoint* Steiner trees in G with terminals K . If n is large enough then the number of rounds needed is $O\left(\frac{n}{T}\right) = O\left(\frac{n}{\lambda_K(G)}\right)$, giving a tight bound.

We now consider the case of $\text{ED}_{K,n}$ with large enough n . The communication bottleneck argument along with known relations between multi-commodity flow and sparsity of a graph [LR99, LLR95] gives us a randomized lower bound of $\tilde{\Omega}(\tau_{\text{MCF}}(G, K, 1))$. The trivial protocol of all players sending (the hash of their) inputs to one player r gives us an upper bound of $\tilde{O}(\tau_{\text{MCF}}(G, K, k))$. The mis-match is because in this case r has an incoming flow of $\Omega(k)$. To avoid this we adapt the argument in [DKO14] to have $\tilde{O}(1)$ phases, where each phase is a more ‘balanced’ multi-commodity flow problem that can be solved in $\tilde{O}(\tau_{\text{MCF}}(G, K, 1))$ number of rounds. The flow problems in these phases are guided by a small circuit that computes the ED function, as in Lemma 3.

It turns out that for the above results for $\text{ED}_{K,n}$ to hold (for randomized complexity), n has to be *exponentially* larger than the size of G , which is not ideal (and something we would like to avoid assuming). It turns out that the reason we need n to be large enough for the above arguments is that the results for Steiner tree packing [Lau07] and those of multicommodity flow [LR99, LLR95] are only proved without any constraints on the diameter of the Steiner trees and the dilation (i.e. the length of the longest flow) of the multicommodity flow. In our arguments, we take both of these factors into account. In particular, for the upper bounds we simply ‘pick’ the best Steiner tree packing and multicommodity flows based on delay constraints.

However, the Steiner-tree packing result of [Lau07] and the flow-cut-gap results of [LR99, LLR95] break down if we impose the diameter constraints on the Steiner-trees, or the dilation constraint on the multi-commodity flow. We need to use other techniques to handle these constraints. For the Steiner-tree packing problem with diameter constraints, we apply the techniques for bi-criteria network design in [MRS⁺98]. In particular, [MRS⁺98] gave an $(O(\log|V|), O(\log|V|))$ -approximation algorithm for the bounded diameter minimum Steiner tree problem, where the first $O(\log|V|)$ factor is for the violation of the diameter constraint and the second $O(\log|V|)$ factor is for the cost of the Steiner tree. Using the duality between maximizing Steiner-tree packing and minimizing cost of a Steiner tree, we are able to give a good Steiner-tree packing that approximately satisfies the diameter constraint.

For the multi-commodity flow problem with the dilation constraint, we could not give a good bi-criteria approximation for all demand functions. However, for certain demand functions (including the demand function corresponding to ED), we can apply the cut-matching-game technique in [KRV09] for constructing expanders. For these demand functions, for every equal partition (A, B) of K , we can route many matchings between A and B using short paths. The cut-matching game technique allows us to find a small-congestion “embedding” of an expander in G using these paths, which along with the properties of the expander allow us to route the demand appropriately.

A crucial ingredient in our lower bound proofs is the notion of a timed graph, which was introduced

in the context of network coding in the study of cyclic networks [ACLY00]. Timed graphs have found several applications in the network coding literature including in the study of time constrained network communication, memory constrained network communication, and *gossip* protocols, e.g., [HKM11, WC14, CKKV15]. Informally, the τ -timed version graph of G , which we denote by $G^{(\tau)}$ is a graph with $\tau + 1$ layers (with $\tau + 1$ copies of V) with the edge set of E repeated between the i and $(i + 1)$ th layer (for $0 \leq i < \tau$). The crucial property of the timed graph is that there exists protocol Π over G with round complexity τ if and only if there exists a protocol on $G^{(\tau)}$ where each edge is used at most once.

Finally, we present an overview of how we argue for the presence of good Steiner tree packing and good multi commodity flow in $G^{(\tau)}$, where τ is the number of rounds taken by the optimal protocol. The obvious thing to try here would be to again appeal to two party communication complexity lower bounds on cuts on $G^{(\tau)}$ itself and then appeal to known results relating Steiner tree packing and multi-commodity flow on directed graphs to the corresponding functions on cuts. There are two issues. First, for Steiner tree packing and multi commodity flow the integrality gap for general directed graphs are either unknown or unbounded (which is not helpful). We get around this issue by explicitly using the fact that $G^{(\tau)}$ is a special graph: i.e. a timed graph of an *undirected* graph. The second issue is that directly applying the two party communication complexity lower bounds across a cut in $G^{(\tau)}$ is typically not enough since these only imply a lower bound on number of crossing edges in *both* directions across a cut in $G^{(\tau)}$, while our argument require a lower bound on the number of edges going from ‘left’ to ‘right’ in a cut. We address this issue by invoking the two party communication complexity not across a cut in $G^{(\tau)}$ but invoking it on a carefully chosen subgraph of $G^{(\tau)}$.

Organization of the paper

We start off with some preliminaries in Section 2. We prove Theorem 4 in Section 3. We prove Theorem 1 (and its generalization Theorem 14) in Section 4. We prove Theorem 2 (and its generalization Theorem 19) in Section 5. Finally, we present our bounds for distributed graph problems in Section 6.

2 Preliminaries

2.1 Notations

Let $f : \{0, 1\}^K \rightarrow \{0, 1\}$ be a function, and $A, B \subseteq K$ be two disjoint non-empty sets and $\tilde{\mathbf{x}} \in \{0, 1\}^{K \setminus (A \cup B)}$. Define $f_{A, B, \tilde{\mathbf{x}}} : \{0, 1\}^A \times \{0, 1\}^B \rightarrow \{0, 1\}$ to be the function such that for every $\mathbf{x}_A \in \{0, 1\}^A$ and $\mathbf{x}_B \in \{0, 1\}^B$, we have $f_{A, B, \tilde{\mathbf{x}}}(\mathbf{x}_A, \mathbf{x}_B) = f(\tilde{\mathbf{x}} \circ \mathbf{x}_A \circ \mathbf{x}_B)$, where $\tilde{\mathbf{x}} \circ \mathbf{x}_A \circ \mathbf{x}_B$ denotes the vector $\mathbf{y} \in \{0, 1\}^K$ such that $\mathbf{y}[v] = \mathbf{x}_A[v]$ if $v \in A$, $\mathbf{y}[v] = \mathbf{x}_B[v]$ if $v \in B$ and $\mathbf{y}[v] = \tilde{\mathbf{x}}[v]$ if $v \in K \setminus (A \cup B)$. For every pair $A, B \subseteq K$ of disjoint non-empty sets, we use $G_{A, B}$ to denote the graph G with vertices in A identified, and vertices in B identified. We shall use v_A and v_B to denote the two new vertices in $G_{A, B}$.

2.2 The timed graph

We now define a graph related to G that will be crucial in our arguments. Given an integer $\tau \geq 1$, we define a directed and layered graph $G^{(\tau)} = (V_\tau, E_\tau)$, where

$$V_\tau = V \times [0, \tau],$$

and E_τ is defined as follows. For every $(u, v) \in E$, we have the following edges in E_τ :

$$\{((u, i), (v, i + 1)) | 0 \leq i < \tau\} \cup \{((v, i), (u, i + 1)) | 0 \leq i < \tau\}.$$

Finally we add to E_τ infinitely many parallel edges $((u, i), (u, i + 1))$ for every $u \in V$ and $0 \leq i < \tau$ that we call *memory edges*.⁶

A useful property of $G^{(\tau)}$ is that given a protocol with congestion c on $G^{(\tau)}$ one can easily construct a protocol with no congestion on $G^{(c \cdot \tau)}$, i.e. we can get a valid protocol on G with delay $c\tau$. This makes our arguments simpler since in the Steiner tree packing and multi-commodity flow solutions we can tolerate $\tilde{O}(1)$ congestion.

2.3 Graph background

We recall two graph problems that have been studied extensively and will be crucial in our analysis.

Steiner Tree Packing. We begin with the problem of Steiner tree packing. Given the graph G and set of terminals K , we call a tree T a Steiner tree if it connects all vertices in K only using edges in G . We consider the (fractional) Steiner tree packing problem, which we will represent by the following well-studied LP. In particular, we would be interested in Steiner trees with diameter (between any two terminals) of Δ —let $\mathcal{T}_{\Delta, K}$ denote the set of all such Steiner trees.

$$\max \sum_{T \in \mathcal{T}_{\Delta, K}} z_T \quad \text{s.t.} \quad \sum_{T \ni e} z_T \leq 1 \text{ for every } e \in E, \quad z_T \geq 0, \forall T \in \mathcal{T}_{\Delta, K}.$$

Let $\text{ST}(G, K, \Delta)$ denote the optimal value of the above LP.

Multi-commodity flow. We will also use the well-studied multi-commodity flow problem. A demand function D is some vector in $\mathbb{R}_{\geq 0}^{K \times K}$. In this demand, we need to send $D_{u,v}$ units of flow from u to v for every $u \in K, v \in K$. Since we are interested in the minimum number of rounds to route the demand function D , it is convenient to view the demand as directed and not necessarily symmetric: for every $u, v \in K$, we need to send $D_{u,v}$ units of flow from u to v and $D_{v,u}$ units of flow from v to u , where $D_{u,v}$ and $D_{v,u}$ may be different. In the problem, we assume that in each round for every edge $(u, v) \in E$, we can send at most 1 unit flow from u to v and at most 1 unit flow from v to u .

Definition 5. For any real number $n' > 0$, we say a demand function $D \in \mathbb{R}_{\geq 0}^{K \times K}$ is n' -bounded, if for every $u \in K$, we have $\sum_{v \in K} D_{u,v} \leq n'$ and $\sum_{v \in K} D_{v,u} \leq n'$.

Definition 6. For every $n' > 0$, let $\tau_{\text{MCF}}(G, K, n')$ be the minimum number of rounds τ such that we can simultaneously send n'/k units flow from u to v in G , for every $u, v \in K$. For every $a, b \in V$, let $\tau_{\text{route}}(G, \{a, b\}, n')$ denote the minimum number of rounds τ such that a can send n' units flow to b .

In other words, $\tau_{\text{MCF}}(G, K, n')$ is minimum number of rounds to route D , for the function D with $D_{u,v} = n'/k$ for every $u, v \in K$. Note that

Proposition 7. When G is a clique on k vertices, we have

$$\tau_{\text{MCF}}(G, K, n') = \left\lceil \frac{n'}{k} \right\rceil.$$

We first note a simple property of τ_{MCF} and τ_{route} .

⁶We only need large enough memory edges. However, we choose to say there are infinite of them to avoid having to specify the exact number of such edges.

Claim 8. For every $n' > 0$ and $n'' > 0$ and $a, b \in V$, we have $\tau_{\text{MCF}}(G, K, n'') \leq \left\lceil \frac{n''}{n'} \right\rceil \tau_{\text{MCF}}(G, K, n')$, and $\tau_{\text{route}}(G, \{a, b\}, n'') \leq \left\lceil \frac{n''}{n'} \right\rceil \tau_{\text{route}}(G, \{a, b\}, n')$.

Next, we note that the definition of τ_{MCF} is enough to capture all n' -bounded demands.

Lemma 9. For every n' -bounded demand D over K , we can route D with $2\tau_{\text{MCF}}(G, K, n')$ rounds.

Proof. Without loss of generality, we assume for every $u \in K$ we have $\sum_{v \in K} D_{u,v} = n'$ and $\sum_{v \in K} D_{v,u} = n'$. We route the demand D in 2 stages, each with delay $\tau_{\text{MCF}}(G, K, n')$. We color the commodities by their destinations. So at the beginning, there are $D_{u,v'}$ units of commodity of color v' at u , for every $u, v' \in K$. In the first stage, we send n'/k units of commodity from every $u \in K$ to every $v \in K$, such that the commodity of each color is split evenly: v is getting $D_{u,v'}/k$ units of commodity of color v' from u , for every color $v' \in K$. Thus, at the end of the first stage, every vertex u has n'/k units commodity of each color v' . Then, in the second stage, we send the commodity of each color v' to v' . Notice that in each of the two stages, we are sending n'/k units of flow from every u to every v and thus the delay is $\tau_{\text{MCF}}(G, K, n')$; so overall the delay is $2\tau_{\text{MCF}}(G, K, n')$. \square

Facts about expanders. Given a graph $H = (V_H, E_H)$, the expansion of H is defined as

$$\Phi(H) := \min_{S \subseteq V_H: |S| \leq |V_H|/2} \frac{|E_H(S, V_H \setminus S)|}{|S|},$$

where $E_H(S, V_H \setminus S)$ is the set of edges in E_H with one endpoint in S and the other endpoint in $V_H \setminus S$. We say a graph is an α -expander if its expansion is at least α .

Let H be a d -regular graph and A be the adjacency matrix of H : for every $u, v \in K$, $A_{u,v}$ is the number of edges between u and v in X . Since A is symmetric, it has n real eigenvalues. The largest eigenvalue of A is $\lambda_1 = d$. Let $\lambda_2 \leq d$ be the second largest eigenvalue of A . Cheeger's inequality relates λ_2 and the expansion $\Phi(H)$ of H .

Theorem 10 (Cheeger's Inequality). $\frac{d-\lambda_2}{2} \leq \Phi(H) \leq \sqrt{2d(d-\lambda_2)}$.

We are interested in the following lazy random walk on a d -regular graph H . We start from an initial vertex $v \in V_H$, chosen randomly according to some initial distribution q . In each step, with probability $1/2$, we stay at the current vertex; with the remaining $1/2$ probability, we move to a randomly selected neighbor of the current vertex. Then, $(I + A/d)/2$ is the transition matrix of the lazy random walk, where I is the identity matrix. The following theorem says that the mixing time of the lazy random walk on an expander is small.

Theorem 11 (Lazy random walk on expanders). Let $H = (V_H, E_H)$ be a d -regular graph with $|V_H| = N_H$, A be its adjacency matrix and λ_2 be the second largest eigenvalue of A . Let $\mu = (\mu_v = \frac{1}{N_H})_{v \in V_H}$ be the uniform distribution over vertices in V_H . For any initial distribution $q \in [0, 1]^{V_H}$ over V_H and integer $T \geq 0$, we have

$$\left\| \left(\frac{I + A/d}{2} \right)^T q - \mu \right\|_1 \leq \sqrt{N_H} \left(\frac{1 + \lambda_2/d}{2} \right)^T.$$

2.4 Circuits

We will consider circuits that compute a function f . In particular, we will consider circuits with gates of fan-out and fan-in at most two: (i) AND, (ii) OR, (iii) NOT and (iv) duplication gate⁷. We will call such a circuit (s, d) -bounded if it has at most s wires and has depth d . In this paper we almost exclusively deal with the case of $d = \tilde{O}(1)$. Also for uniformity, we will think of each input bit as a ‘constant gate.’

3 The case of $k = 2$

In this section we consider the special case of $K = \{a, b\}$ (for Alice and Bob) but still over an arbitrary graph G . Our main result is Theorem 4, which we prove in this section and is re-stated below:

Theorem 4 (Restated). *For any function $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, and any graph G we have that*

$$\tau_{\text{route}}(G, \{a, b\}, R_\epsilon^{(2)}(f)) \leq 4R_\epsilon(f, G, \{a, b\}).$$

We would again like to stress that our proof does not proceed by invoking two party communication complexity lower bounds on two party functions induced by cuts on G . However, we do prove the result via a two-party communication simulation (where the two parties are denoted by a' and b'). We will argue that if the result is not true, then we can come up with a two-party protocol for f with cost strictly less than $R_\epsilon^{(2)}(f)$, which would lead to a contradiction. Roughly speaking, if we can not route $R_\epsilon^{(2)}(f)/2$ units flow from a to b in time $2\tau := 2R_\epsilon(f, G, \{a, b\})$, then we can remove a few special edges in G to make the distance between a and b to be at least 2τ (see Lemma 12). Then we can divide the vertices in G into $2\tau + 1$ levels indexed from 0 to 2τ , such that a is at level 0, b is at level 2τ and all non-special edges are between two adjacent levels or between vertices in the same level. Thus, the flow of information via non-special edges is slow; in particular, without the special edges, the information from a and b will not mix in τ rounds. Informally speaking, the important bits in the protocol are those sent via special edges. Thus, in the simulation using a two-party protocol between a' and b' , we only send these important bits. By a careful analysis, we can bound the number of bits sent between a' and b' by less than $R_\epsilon^{(2)}(f)$, leading to a contradiction.

Proof of Theorem 4. It would be convenient to consider the set \vec{E} of directed edges, obtained from E by replacing each edge $(u, v) \in E$ with two directed edges (u, v) and (v, u) . Consider the protocol in graph G that computes the function f in $\tau := R_\epsilon(f, G, \{a, b\})$ steps. For every $(u, v) \in \vec{E}$, let $x_{u,v}^t$ be the bit sent from u to v at time t (recall that we allow both directions of an edge in E to be used simultaneously). The bit $x_{u,v}^t$ is a function of the bits received by u by time $t - 1$ (and the public random string); here we assume that a received the input string \mathbf{x}_a and b received the input string \mathbf{x}_b at time 0. We assume towards the contradiction that $\tau_{\text{route}}(G, \{a, b\}, R_\epsilon^{(2)}(f)) > 4\tau$. By Claim 8, we have $\tau_{\text{route}}(G, \{a, b\}, N) > 2\tau$, where $N = \lceil R_\epsilon^{(2)}(f)/2 \rceil$. This says that one cannot route N bits in 2τ rounds from a to b .

Lemma 12. *Given a graph $G = (V, E)$ and $a, b \in V$, assume there is no protocol that sends N bits from a to b in T rounds. Then there exists a vector $\ell \in \{0, 1, 2, \dots, T + 1\}^V$ such that $\ell_a = 0, \ell_b = T + 1$ and*

$$\sum_{(u,v) \in E} \max\{|\ell_u - \ell_v| - 1, 0\} < N.$$

⁷This gate takes one bit as input and outputs two copies of the input bit.

Proof. We consider the time graph $G^{(T)}$. Since there is no protocol that sends N bits from a to b in time T , we can not send N units of flow from $(a, 0)$ to (b, T) in $G^{(T)}$ (with congestion 1). By the max-flow-min-cut theorem, there is a cut of size strictly smaller than N in $G^{(T)}$ that separates $(a, 0)$ from (b, T) . Let (A, B) be the cut in $G^{(T)}$. For every $t \in \{0, 1, 2, \dots, T\}$, let $A_t = \{v \in V : (v, t) \in A\}$. Since there are infinitely many memory edges $((v, t), (v, t+1))$, no such edge can be cut, and we have that $a \in A_0 \subseteq A_1 \subseteq A_2 \subseteq \dots \subseteq A_T \not\supseteq b$. Now, the (A, B) cut value is exactly

$$\sum_{t=0}^{T-1} \sum_{u \in A_t} \sum_{v \notin A_{t+1}} \mathbf{1}_{(u,v) \in E} < N.$$

For each $t \in 1, 2, 3, \dots, T$, we define $V_t = A_t \setminus A_{t-1}$. Define $V_0 = A_0$ and $V_{T+1} = V \setminus A_T$. Thus, $a \in V_0$ and $b \in V_{T+1}$ and $(V_0, V_1, V_2, \dots, V_{T+1})$ forms a partition of V . For each $v \in V$, let ℓ_v be the index such that $v \in V_{\ell_v}$. We claim that in the above sum, each $(u, v) \in E$ is counted exactly $\max\{0, |\ell_u - \ell_v| - 1\}$ times. Without loss of generality assume $\ell_u \leq \ell_v$. Then, $u \in A_t$ and $v \notin A_{t+1}$ for $\ell_u \leq t < \ell_v - 1$. Thus, (u, v) is counted exactly $\max\{|\ell_u - \ell_v| - 1, 0\}$ times. Thus, we have

$$\sum_{(u,v) \in E} \max\{|\ell_u - \ell_v| - 1, 0\} < N,$$

which concludes our assertion. \square

Applying Lemma 12 with $T = 2\tau$, we obtain vector $\ell \in \{0, 1, 2, \dots, 2\tau + 1\}^V$ satisfying the properties stated in the lemma. We shall use a two-party protocol to simulate the protocol on G ; we use a' and b' to denote the two parties participating in the two-party protocol. We assume a' knows the input string \mathbf{x}_a and b' knows the input string \mathbf{x}_b . The two-party protocol has τ rounds that correspond to the τ time steps of the protocol on G and is defined as follows. In each round t for $t = 1$ to τ , for each bit $x_{u,v}^t$ sent from u to v in the original protocol on G , $x_{u,v}^t$ is either sent from a' to b' , or from b' to a' , or not sent at all according to the following conditions:

- If $\ell_u < t < \ell_v$ then bit $x_{u,v}^t$ is sent from a' to b' in the two-party protocol.
- If $\ell_v < 2\tau + 1 - t < \ell_u$ then bit $x_{u,v}^t$ is sent from b' to a' in the two-party protocol.
- Otherwise bit $x_{u,v}^t$ is not sent in the two-party protocol.

We now claim that (i) the number of bits sent in the two-party protocol is at most $2N - 2$ (where recall $N = \lceil R_\epsilon^{(2)}(f)/2 \rceil$), and (ii) the protocol is valid in the sense that the two parties can compute the bits transmitted during its execution, and once completed, both parties a' and b' know the output of the graph protocol. The fact that the round complexity is bounded by $2N - 2$ follows directly by our definitions. Namely, in the two-party protocol, for any edge $e = (u, v) \in E$, there are at most $2(|\ell_u - \ell_v| - 1)$ different t 's for which the bit $x_{u,v}^t$ or $x_{v,u}^t$ is sent between a' and b' . As by Lemma 12, $\sum_{(u,v) \in E} \max\{|\ell_u - \ell_v| - 1, 0\} \leq N - 1$ we conclude (i) above. We now prove the validity of the protocol.

Lemma 13. *Let $t \in [0, \tau]$. (1) At the end of round t : if $\ell_v \leq 2\tau - t$, then a' knows all the bits received by v ; if $\ell_v \geq t + 1$, then b' knows all the bits received by v . (2) If $t < \tau$ then in round $t + 1$, a' knows all the bits she needs to send to b' , and b' knows all the bits he needs to send to a' .*

Proof. We first show that for each $t \in [0, \tau - 1]$, (1) implies (2). If a' needs to send $x_{u,v}^{t+1}$ to b' in round $t + 1$, then we must have $\ell_u < t + 1$. $x_{u,v}^{t+1}$ depends on all the bits received by u by the end of round t . Since $\ell_u < t + 1 < 2\tau - t$, (1) implies that a' knows all these bits and thus can compute $x_{u,v}^{t+1}$. Thus, a' knows all the bits she needs to send to b' in round $t + 1$; similarly, b' knows all the bits he needs to send to a' .

We now prove the lemma by induction on t ; for each t we only need to prove (1). The base case is $t = 0$; at the end of round 0, a' knows all the bits received by v if $v \neq b$ and b' knows all the bits received by v if $v \neq a$. So, (1) holds since $\ell_a = 0$ and $\ell_b = 2\tau + 1$.

Consider some $t \geq 1$ and assume (1) holds for $t - 1$. We prove (1) for t ; we only need to prove the statement for b' , since the statement for a' can be proved symmetrically. Let $\ell_v \geq t + 1$ and we need to prove that b' knows all the bits received by v before the end of round t . Since $\ell_v \geq (t - 1) + 1$, by the induction hypothesis, b' knows all the bits received by v before the end of round $t - 1$. We only need to show that b' knows all the bits received by v at round t .

Focus on a vertex u such that $(u, v) \in \vec{E}$. In the graph protocol, the bit $x_{u,v}^t$ is sent from u to v at time t . We consider two cases. First consider the case that $\ell_u \geq t$. Thus $\ell_u \geq (t - 1) + 1$; by the induction hypothesis, b' knows all the bits received by u before the end of round $t - 1$; thus b' can compute $x_{u,v}^t$. For the other case, we have $\ell_u < t < \ell_v$. In the two-party protocol, a' sends $x_{u,v}^t$ to b' , implying that b' knows $x_{u,v}^t$ by the end of round t (notice that by induction hypothesis, (2) holds for $t - 1$; thus a' knows $x_{u,v}^t$). This finishes the proof of the lemma. \square

Lemma 13 implies point (ii) above. Indeed, at the end of round τ , a knows the output; a' knows all bits received by a as $\ell_a = 0 \leq 2\tau - \tau$. So a' knows the output. Similarly b' knows the output. Notice that $2N - 2 < R_\epsilon^{(2)}(f)$. The error of the two-party protocol on every input $(\mathbf{x}_a, \mathbf{x}_b)$ is exactly the same as the error of the graph protocol on this input. Thus, we obtain a two-party protocol with total communication less than $R_\epsilon^{(2)}(f)$ and error ϵ ; this contradicts the definition of $R_\epsilon^{(2)}(f)$. So the theorem holds. \square

4 Steiner Tree Packing Bounds

In this section we consider general sets K . We first present a lower bound on $R_\epsilon(f, G, K)$ based on the notion of Steiner tree packing. We then explore the potential optimality of conceptually simple protocols that perform computation of f over a (collection of) Steiner trees that span K .

We prove the following general lower bound result:

Theorem 14. *Let $G, K, f : \{0, 1\}^n \rightarrow \{0, 1\}$, $\epsilon \geq 0$ be defined as usual. Assume for some $n' > 0$ that the following is true: for every pair of distinct players $a, b \in K$, there exists some $\tilde{\mathbf{x}} \in \{0, 1\}^{K \setminus \{a, b\}}$, such that $R_\epsilon^{(2)}(f_{\{a\}, \{b\}, \tilde{\mathbf{x}}}) \geq n'$. Then,*

$$\min_{\Delta \in [|V|]} \left(\frac{n'}{\text{ST}(G, K, \Delta)} + \Delta \right) \leq \tilde{O}\left(R_\epsilon(f, G, K)\right).$$

We provide an overview of the proof of the above result for the case of $\text{DISJ}_{K, n}$. We first use Theorem 4 to get many edge disjoint paths between every pair of terminals in K with length at most $\tilde{O}(\tau)$, where the optimal protocol takes τ rounds. (This follows from the fact that $\tau_{\text{route}}(G, \{a, b\}, R_\epsilon^{(2)}(f')) \leq 4\tau$ via Theorem 4.) Then using tools developed in earlier work on packing Steiner trees with bounded diameter by Marathe et al. [MRS⁺98], we show that we can stitch these sets of edge disjoint path to obtain a large enough set of edge disjoint Steiner tree packings with diameter $\tilde{O}(\tau)$. This is enough to prove our lower bound for $\text{DISJ}_{K, n}$. Next, we prove the result for general f .

Proof of Theorem 14. Assume that $R_\epsilon(f, G, K) = \tau$. In particular, for any a and b in K it holds that $R_\epsilon(f_{\{a\}, \{b\}, \tilde{\mathbf{x}}}, G, \{a, b\}) \leq \tau$. By Theorem 4, for $f' = f_{\{a\}, \{b\}, \tilde{\mathbf{x}}}$, and the fact that $n' \leq R_\epsilon^{(2)}(f')$,

$$\tau_{\text{route}}(G, \{a, b\}, n') \leq \tau_{\text{route}}(G, \{a, b\}, R_\epsilon^{(2)}(f')) \leq 4R_\epsilon(f', G, \{a, b\}) \leq 4\tau.$$

Thus, there exists n' edge-disjoint paths connecting $(a, 0)$ and $(b, 4\tau)$ in $G^{(4\tau)}$. This in turn implies n' fractional edge disjoint paths in G of length at most 4τ in which each path has fractional value $\frac{1}{4\tau}$, yielding a total value of $\frac{n'}{4\tau}$ fractional edge disjoint paths (of length at most 4τ). The analysis above holds for all pairs a and b in K . In what follows (in Theorem 15 given below), we show that the latter implies a fractional Steiner Tree packing in G of value $\tilde{\Omega}\left(\frac{n'}{\tau}\right)$ with tree diameter at most $\tilde{O}(\tau)$. Implying that:

$$\left(\min_{\Delta \in [|V|]} \left(\frac{n'}{\text{ST}(G, K, \Delta)} + \Delta \right) \right) \leq \tilde{O}(\tau) = \tilde{O}(R_\epsilon(f, G, K)).$$

□

We now address the missing assertion in the proof of Theorem 14. We start with some notation. Given a (partial) matching M over K and a set \mathcal{P} of $|M|$ edge-disjoint paths in G , we say \mathcal{P} supports M if for every $(a, b) \in M$, there is a path in \mathcal{P} connecting a and b . We prove that

Theorem 15. *Let $K = \{u_0, u_1, \dots, u_{k-1}\}$. Assume that for every $u_i \in K \setminus \{u_0\}$ there is a collection \mathcal{Q}_i of fractional edge-disjoint paths of length at most D from u_i to u_0 in G with total value p . Then, there is a Steiner tree packing of value $\tilde{\Omega}(p)$ in G with tree diameter at most $\tilde{O}(D)$.*

Proof. We use the following lemma:

Lemma 16. *There is a randomized algorithm that given $K' \subseteq K$ of even cardinality outputs a matching M over K' and a set \mathcal{P} of $|M|$ edge disjoint paths supporting M such that (i) $|M| \geq |K'|/4$, (ii) all paths in \mathcal{P} have length at most $16D$, and (iii) for every $e \in E$, $\Pr[e \text{ is used by paths in } \mathcal{P}] \leq 4/p$.*

Proof. Let E' be the set of all edges used by paths in $\cup_{u_i \in K'} \mathcal{Q}_i$, let $w_e \leq 1$ be the total weight of paths in \mathcal{Q}_i that use e , and let $w(E')$ be the sum of edge weights of edges in E' . So $w(E') \leq |K'|pD$. Let $G' = (V, E')$ with edge capacities w_e . By [LL04] we can find a fractional Steiner tree packing (\mathcal{T}', z') of value $p/2$ in G' . However, there is no guarantee for the diameters of the trees in \mathcal{T}' . Focus on each tree $T \in \mathcal{T}'$. It is not hard to find a perfect matching M over K' , and a set of $|M|$ edge-disjoint paths \mathcal{P} in T that supports M . Here, one needs to pair the elements of K' iteratively starting from the pair with the least common ancestor which is furthest from a predefined root, removing that pair, and recursing. We say a path $P \in \mathcal{P}$ is short if its length is at most $16D$; otherwise, we say P is long. We say that T is bad if the number of long paths in \mathcal{P} is at least $|K'|/4$; otherwise, we say T is good. It follows that $\sum_{T \in \mathcal{T}': T \text{ bad}} z'_T \leq p/4$, as otherwise we have $w(E') > 16D \times |K'|/4 \times p/4 = |K'|pD$. A contradiction. Thus, $Z' = \sum_{T \in \mathcal{T}': T \text{ good}} z'_T \geq p/4$.

The randomized algorithm now works as follows. We first randomly choose a good tree $T \in \mathcal{T}'$ with probability z'_T/Z' . Then we take the perfect matching M over K' and the set of $|M|$ edge-disjoint paths \mathcal{P} in T that support M . We remove all long paths from \mathcal{P} and their corresponding pairs from M . Then we output (M, \mathcal{P}) . As each edge $e \in E'$ has $w_e \leq 1$, we have that

$$\Pr[e \text{ is used by paths in } \mathcal{P}] = \frac{\sum_{T \in \mathcal{T}': T \text{ good}, T \ni e} z'_T}{Z'} \leq \frac{w_e}{Z'} \leq \frac{4}{p}.$$

This finishes the proof of Lemma 16. □

We now proceed to the proof of Theorem 15. We shall define a randomized algorithm to output a Steiner tree T over K of diameter at most $\tilde{O}(D)$. The final packing is implicitly defined by the randomized algorithm. That is, a tree T has z_T value proportional to the probability that the randomized algorithm outputs T . The algorithm is a simple application of Lemma 16 above and proceeds as follows: Initially,

set $K' \leftarrow K$, $T \leftarrow \emptyset$. Now, repeat the following steps until $|K'| = 1$: (i) apply Lemma 16 to find a matching M over K' and its corresponding supporting paths \mathcal{P} , (ii) add the edges in \mathcal{P} to T , and (iii) for every $(u, v) \in M$, arbitrarily remove one of the two vertices in $\{u, v\}$ from K' . Finally, return T . Note that this procedure recurses $\lceil \log_{4/3} k \rceil \leq 4 \log k$ many times (and the final diameter and congestion in the worst-case gets multiplied by $4 \log k$). Lemma 16 implies that the diameter of T is at most $64D \log k = \tilde{O}(D)$. Moreover, for every $e \in E$, the probability that $e \in T$ is at most $\frac{16}{p} \log k$.

To obtain the fractional Steiner Tree packing, let p_T be the probability that tree T is returned by the randomized algorithm. It follows that $z_T = \frac{p}{16 \log k} p_T$ is a solution to the Steiner Tree packing LP of value $\frac{p}{16 \log k} = \tilde{\Omega}(p)$. This finishes the proof of Theorem 15. \square

4.1 Steiner tree upper bounds

We consider a reasonably large class of composed functions. In particular, given a function $g : \{0, 1\}^n \rightarrow \{0, 1\}$, the class of functions $g \circ \text{SYMM}$ is the class of all functions $f : \{0, 1\}^K \rightarrow \{0, 1\}$ such that there exists ‘inner’ symmetric functions $h_i : \{0, 1\}^K \rightarrow \{0, 1\}$ for $i \in [n]$ such that

$$f(\{\mathbf{x}_u\}_{u \in K}) = g(h_1(\{\mathbf{x}_u[1]\}_{u \in K}), \dots, h_n(\{\mathbf{x}_u[n]\}_{u \in K})).$$

Note that $\text{DISJ}_{K,n}$ is a special case when g is the n -bit OR and h_i is the k -bits AND. Next, we argue that all such functions have a simple Steiner tree type upper bound. We now show that

Lemma 17. *For any graph G and subset of players K , let f be in $g \circ \text{SYMM}$ for an arbitrary $g : \{0, 1\}^n \rightarrow \{0, 1\}$. Then*

$$R_0(f, G, K) \leq \tilde{O} \left(\min_{\Delta \in [|V|]} \left(\frac{n}{\text{ST}(G, K, \Delta)} + \Delta \right) \right).$$

Proof. Let $\Delta \in [|V|]$ and consider an optimal fractional solution to the Δ -diameter Steiner Tree (ST) packing LP of value $\text{ST}(G, K, \Delta)$. Such a solution can be rounded to an integral ST packing of value $\tilde{\Omega}(\text{ST}(G, K, \Delta))$ [RT87]. Let $u_0 \in K$. For every tree in the ST packing it is straightforward to schedule the transmission of a stream of bits from each terminal in $K \setminus \{u_0\}$ towards u_0 such that vertex u_0 receives for all $i \in [n]$, the sum of the i th bits in $K \setminus \{u_0\}$. Note that since the h_i ’s are symmetric functions this is enough for u_0 to compute the value of f . If each terminal u holds m bits $\mathbf{x}_u \in \{0, 1\}^m$, using a single tree, vertex u_0 will be able to compute the sum of the collection $\{\mathbf{x}_u \in \{0, 1\}^m\}_{u \in K}$ in at most $m \lceil \log k \rceil + \Delta$ rounds. Using the $\tilde{\Omega}(\text{ST}(G, K, \Delta))$ trees in parallel one may set $m = \tilde{O} \left(\frac{n}{\text{ST}(G, K, \Delta)} \right)$ on each tree to conclude our assertion. \square

4.2 Some tight bounds

As noted earlier, $\text{DISJ}_{K,n}$ is a special case of the composed function from Section 4.1. Lemma 17 along with Theorem 14 (where we use the well-known lower bounds for two-party DISJ [Raz92] and setting $\tilde{\mathbf{x}}$ to be the all 1s vector) proves Theorem 1.

We sketch how this result can be extended to a larger family of composed functions.

Proposition 18. *Consider the class of all composed functions (in the sense of Section 4.1) where all the inner symmetric functions h_i ’s are not the constant function, the parity (or its negation). Further, the outer function is such that $g(\mathbf{x}[1] \vee \mathbf{y}[1], \dots, \mathbf{x}[n] \vee \mathbf{y}[n])$ has two party-communication complexity of $\Omega(n)$. Then for any $\epsilon \geq 0$, every function f in this class satisfies:*

$$R_\epsilon(f, G, K) = \tilde{\Theta} \left(\min_{\Delta \in [|V|]} \left(\frac{n}{\text{ST}(G, K, \Delta)} + \Delta \right) \right).$$

Note that $\neg\text{DISJ}_{K,n}$ belongs to this class of functions.

Proof Sketch of Proposition 18. Lemma 17 proves an upper bound of $\tilde{O}\left(\min_{\Delta \in [|V|]} \left(\frac{n}{\text{ST}(G,K,\Delta)} + \Delta\right)\right)$. Further, since the h_i 's are not one of the four ruled out functions, there is always a way to fix any $k-2$ of the inputs (other than say the terminals a and b) such that value of f is determined by $g'(\mathbf{x}_a, \mathbf{x}_b) = g(\mathbf{x}_a[1] \vee \mathbf{x}_b[1], \dots, \mathbf{x}_a[n] \vee \mathbf{x}_b[n])$. Indeed, by the choice of h_i , for every $i \in [n]$, there exist a value $0 \leq c_i < k-1$ such that h_i evaluates to different values on inputs with c_i and $c_i + 1$ ones. Further, it evaluates to the same value on inputs of size $c_i + 1$ and $c_i + 2$. In other words, if we pick $\tilde{\mathbf{x}}$ such that the sum of the number of ones among $\tilde{\mathbf{x}}[u]$ for all $u \in K \setminus \{a, b\}$ in the i th position is exactly c_i , then we note that $f_{\{a\}, \{b\}, \tilde{\mathbf{x}}}$ is exactly $g(\mathbf{x}_a[1] \vee \mathbf{x}_b[1], \dots, \mathbf{x}_a[n] \vee \mathbf{x}_b[n])$. By assumption g' has $\Omega(n)$ two party communication complexity, which by Theorem 14 implies an overall lower bound of $\tilde{\Omega}\left(\min_{\Delta \in [|V|]} \left(\frac{n}{\text{ST}(G,K,\Delta)} + \Delta\right)\right)$. \square

5 Multicommodity flow type bounds

5.1 Circuits to Protocols

Here we sketch the proof of Lemma 3, which we re-state below:

Lemma 3 (Restated). *Let $f : (\{0, 1\}^n)^k \rightarrow \{0, 1\}$ have a circuit with constant fan-in and constant fan-out gates and depth d . Further, each level $i \in [d]$ has s_i gates in it. Then*

$$R_0(f, G, K) \leq \sum_{i=1}^d \tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{s_i}{k}\right)\right). \quad (1)$$

Finally, we can upper bound the above by $\tilde{O}(d \cdot \tau_{\text{MCF}}(G, K, \frac{s}{k}))$ as well as $\tilde{O}(\frac{s}{k} \cdot \tau_{\text{MCF}}(G, K, 1))$.

The proof is an adaptation of an idea that was used in [DKO14] to design protocols for G being a clique (i.e. the CONGEST-CLIQUE model). Let C be the given circuit for f . Then one can assign each gate of C to each terminal in K and then we evaluate each layer by setting up a multi-commodity flow problem where for each gate g in the current level all the input gates (or their assigned terminals) send their value to g (or the player that is assigned to g). Since at level i C has s_i gates, it can be shown via the probabilistic method that there exists an assignment of gates such that each terminal only has a total requirement of $\tilde{O}(s_i/k)$. We now present the details.

Proof of Lemma 3. Assuming (1) is correct, we note that the second bound follows by the simple observation that $s_i \leq s$. Further, the third bound follows from Claim 8 and the fact that $\sum_{i=0}^d s_i \leq s$.

We now argue (1). Let C be the given (s, d) -bounded circuit for f . For every $0 \leq i \leq d$, let s_i be the number of gates a level i . (Note that $s_0 = nk$ and $s_d = 1$.) The idea is to evaluate the circuit C in the given delay. We will do so by evaluating all gates in a given level one at a time. In particular, we will argue that we can evaluate the gates at level i with delay $\tilde{O}(\tau_{\text{MCF}}(G, K, \frac{s_i}{k}))$. Note that this suffices to prove (1).

We will need a map from the gates of C to terminals in K with certain properties. To show the existence of such a map, let π denote a random map from the s gates of C to the k players. We note that by a standard balls and bins argument, any set of $\Theta(s_i)$ gates are assigned to any specific player with load $L_i = O(\frac{s_i}{k} \log kd)$ with probability $> 1 - 1/(2d)$. (We will see shortly that this is enough to handle all bad cases that may arise in the rest of our arguments.) We begin with level 0. Note that in this case the $s_0 = nk$ input bits would need to be re-routed according to π . By the balls and bins argument, this means we have a demand set where each player has load $L_0 + s_0/k$. (Recall that initially each player has $n = s_0/k$ bits.)

Thus, we can ‘evaluate’ level 0 with delay $\tau_{\text{MCF}}(G, K, L_0 + s_0/k)$, which by Claim 8 is $\tilde{O}(\tau_{\text{MCF}}(G, K, \frac{s_0}{k}))$, as desired.

Assume by induction we have evaluated all levels up to level $i \geq 0$. Now consider level $i + 1$. Consider an arbitrary gate g whose inputs are gates g' (and possibly) g'' . We add a demand pair with requirement 1 between the pairs $(\pi(g), \pi(g'))$ and $(\pi(g), \pi(g''))$. Note that since there are s_{i+1} such gates g and at most $2s_{i+1}$ input gates from previous levels. Thus, by the balls and bins argument, each player has at most $3L_{i+1}$ of the gates at level $i + 1$ and their inputs. This implies that $\tau_{\text{MCF}}(G, K, 3L_{i+1})$ rounds suffice to evaluate level $i + 1$, which by Claim 8 is $\tilde{O}(\tau_{\text{MCF}}(G, K, \frac{s_{i+1}}{k}))$, as desired.

Finally we note that we had at most $2d - 1$ bad events (where a bad event is at level i some player has more than L_i gates from level i or one of its input gates assigned to it) that we would like π to avoid. By the union bound, there exists a map π that makes the protocol above go through with the required round complexity. \square

5.2 The lower bound

We are now ready to state our most general lower bound.

Theorem 19. *Let $G, K, f : (\{0, 1\}^n)^K \rightarrow \{0, 1\}, \epsilon \geq 0$ be defined as usual, and assume k is even. Let $h : [k/2] \times [k/2] \rightarrow \mathbb{R}_{\geq 0}$. Assume the following is true: for every pair of disjoint sets $A, B \subseteq K$ such that $|A|, |B| \leq k/2$, there exists some $\tilde{\mathbf{x}} \in (\{0, 1\}^n)^{K \setminus (A \cup B)}$, such that $R_\epsilon^{(2)}(f_{A, B, \tilde{\mathbf{x}}}) \geq h(|A|, |B|)$. Then,*

$$\tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R_\epsilon(f, G, K)), \quad (2)$$

where $n' = \min_{y, z \in [0, k/2]: y+z > k/2} \frac{h(y, z)}{y+z-k/2}$.

The above implies lower bounds for the $\text{ED}_{K, n}$ function:

Corollary 20. *For any G and K , if $n \geq 1 + 2 \lceil \log k \rceil$, then we have*

$$\tau_{\text{MCF}}(G, K, 1) \leq \tilde{O}(R(\text{ED}_{K, n}, G, K)), \quad \text{and} \quad \tau_{\text{MCF}}(G, K, n) \leq \tilde{O}(R_0(\text{ED}_{K, n}, G, K)).$$

Proof. Let $f = \text{ED}_{K, n}$. Fix some $A, B \subseteq K$ such that $A \cap B = \emptyset$ and $|A| \leq |B| \leq k/2$. We shall let $\tilde{\mathbf{x}} \in (\{0, 1\}^n)^{K \setminus (A \cup B)}$ be a vector so that $\tilde{\mathbf{x}}_{v,1} = 1$ for every $v \in K \setminus (A \cup B)$, and the $|K \setminus (A \cup B)|$ vectors $\{\tilde{\mathbf{x}}_v\}_{v \in K \setminus (A \cup B)}$ are different. This is possible since $n \geq 1 + 2 \lceil \log k \rceil$. Then for the function $f_{\tilde{\mathbf{x}}, A, B}(\mathbf{x}_A, \mathbf{x}_B)$, we are interested in the input pairs $(\mathbf{x}_A, \mathbf{x}_B)$ such that $\mathbf{x}_{A, v}[1] = 0$ for every $v \in A$ and $\mathbf{x}_{B, v}[1] = 0$ for every $v \in B$. Thus, $f_{\tilde{\mathbf{x}}, A, B}(\mathbf{x}_A, \mathbf{x}_B) = 1$ if and only if the $|A| + |B|$ strings $\{\mathbf{x}_A[v]\}_{v \in A} \cup \{\mathbf{x}_B[v]\}_{v \in B}$ are all different. In other words, we want to compute the two party DISJ problem on the sets $\{\mathbf{x}_A[v]\}_{v \in A}$ and $\{\mathbf{x}_B[v]\}_{v \in B}$. It is well-known that $R^{(2)}(f_{A, B, \tilde{\mathbf{x}}}) \geq \Omega(|A|)$ ([HW07]). We argue from first principles in Theorem 38 that $R_0^{(2)}(f_{A, B, \tilde{\mathbf{x}}}) \geq \Omega(n|A|)$.

Let $\epsilon = 1/3$. Let $n' = \tilde{\Omega}(1)$ be small enough. Let $h(y, z) = n' \min\{y, z\}$ for every $y, z \in [k/2]$. Then $\min_{y, z \in [k/2]: y+z > k/2} \frac{h(y, z)}{y+z-k/2} = \min_{0 \leq y \leq z \leq k/2: y+z > k/2} \frac{yn'}{y+z-k/2} = n'$. Thus, if n' is small enough, then the condition for Theorem 19 holds. Thus, we have

$$\tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R(\text{ED}_{K, n}, G, K)).$$

Then by Claim 8, $\tau_{\text{MCF}}(G, K, 1) \leq \lceil \frac{1}{n'} \rceil \tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(1) \tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R(\text{ED}_{K, n}, G, K))$.

Let $\epsilon = 0$. Let $n' = \Omega(n)$ be small enough. Let $h(y, z) = n' \min\{y, z\}$ for every $y, z \in [k/2]$. Again, if n' is small enough, then the condition for Theorem 19 holds. Thus, we have

$$\tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R_0(\text{ED}_K, G, K, n)).$$

Then $\tau_{\text{MCF}}(G, K, n) \leq \lceil \frac{n}{n'} \rceil \tau_{\text{MCF}}(G, K, n') \leq O(1) \tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R_0(\text{ED}_{K, n}, G, K))$, by Claim 8. \square

In Section 6 we make use of other corollaries of Theorem 19.

We now sketch the proof of Theorem 19 (specialized to $R(\text{ED}_{K,n}, G, K)$). First we note that any cut separating k' terminals from the rest of the $k - k'$ terminals induces a two party communication complexity problem that needs $\Omega(\min(k', k - k'))$ communication across the cut. This in conjunction with our argument for $k = 2$ implies that there are $\Omega(\min(k', k - k'))$ edge disjoint paths between the two subsets in $G^{(\tau)}$. We now use the cut-matching game framework of Khandekar, Rao and Vazirani [KRV09] to argue that we can construct an expander graph each of whose edges can be embedded into $G^{(\tau)}$ in the sense that each edge in the expander corresponds to a path in $G^{(\tau)}$ (and these paths have low congestion). Since the multicommodity flow with a total demand requirement of $\tilde{O}(k)$ from each terminal can be done with $d = \tilde{O}(1)$ delay on the expander graph, we can route these paths in $G^{(d \cdot \tau)}$. (We need to make sure that the paths in the expander are not too long but this can be done.) This implies a protocol for the multi-commodity flow problem that we need to solve for the upper bound with delay $\tilde{O}(\tau)$, as desired.

We now formally prove Theorem 19. Fix any two disjoint sets $A, B \subseteq K$ such that $|A|, |B| \leq k/2$. Let $\tilde{\mathbf{x}} \in (\{0, 1\}^n)^{K \setminus (A \cup B)}$ be the vector satisfying the condition of the theorem for the pair (A, B) .

By Theorem 4, we have that

$$\tau_{\text{route}}(G_{A,B}, \{v_A, v_B\}, R_\epsilon^{(2)}(f_{A,B}, \tilde{\mathbf{x}})) \leq 4R_\epsilon(f_{A,B}, \tilde{\mathbf{x}}, G, \{v_A, v_B\}).$$

where we overload notation for R_ϵ and $R_\epsilon^{(2)}$ by allowing input function to have inputs from different domains: i.e. unlike the original definition, $f_{A,B}, \tilde{\mathbf{x}} : (\{0, 1\}^n)^A \times (\{0, 1\}^n)^B$ has two inputs from different domains.⁸

It is easy to see that $R_\epsilon(f_{A,B}, \tilde{\mathbf{x}}, G_{A,B}, \{v_A, v_B\}) \leq R_\epsilon(f, G, K)$, since every protocol to compute f in G among K leads to a protocol to compute $f_{A,B}, \tilde{\mathbf{x}}$ in $G_{A,B}$ between v_A and v_B . Let $\tau = \lceil 4R_\epsilon(f, G, K) \rceil$. Then, $\tau_{\text{route}}(G_{A,B}, \{v_A, v_B\}, R_\epsilon^{(2)}(f_{A,B}, \tilde{\mathbf{x}})) \leq \tau$.

Since $R_\epsilon^{(2)}(f_{A,B}, \tilde{\mathbf{x}}) \geq h(|A|, |B|)$, we have $\tau_{\text{route}}(G_{A,B}, v_A, v_B, h(|A|, |B|)) \leq \tau$. Thus, there are $h(|A|, |B|)$ edge disjoint paths from $(v_A, 0)$ to (v_B, τ) in $G_{A,B}^\tau$. This implies that there are $h(|A|, |B|)$ edge-disjoint paths from $A \times \{0\}$ to $B \times \{\tau\}$ in G^τ . To see this, focus on each of the $h(|A|, |B|)$ edge-disjoint paths from $(v_A, 0)$ to (v_B, τ) in $G_{A,B}^\tau$. Let t be the smallest number such that (v_B, t) is in the path; let t' be the largest number such that $t' < t$ and (v_A, t') is in the path. Then, we modify this path as follows: we travel from $(v_A, 0)$ to (v_A, t') using memory edges and then use the segment of the path from (v_A, t') to (v_B, t) , and then travel from (v_B, t) to (v_B, τ) using the memory edges. After the modifications, the $h(|A|, |B|)$ edge-disjoint paths in $G_{A,B}^\tau$ can be naturally mapped back to $h(|A|, |B|)$ edge-disjoint paths in G^τ from $A \times \{0\}$ to $B \times \{\tau\}$. Next, we argue that these paths have even more structure.

Lemma 21. *For partition (A, B) of K such that $|A| = |B| = k/2$, we can find $n'k/2$ edge-disjoint paths from $A \times \{0\}$ to $B \times \{\tau\}$ in G^τ , such that every vertex in $A \times \{0\}$ is the origin of exactly n' paths, and every vertex in $B \times \{\tau\}$ is the destination of exactly n' paths.*

Proof. Construct a directed graph \tilde{G} as follows. We start from G^τ , and add a super source s and a super sink t . Then for every $u \in A$, we add n' edges from s to $(u, 0)$. For every $v \in B$, we add n' edges from (v, τ) to t . To prove the lemma, it suffices to show that there are $n'k/2$ edge-disjoint paths from s to t in \tilde{G} . Assume otherwise. Then, there is an s - t cut (S, T) in \tilde{G} whose size is strictly less than $n'k/2$. Let A' be the subset of A such that $S \cap (A \times \{0\}) = A' \times \{0\}$; let B' be the subset of B such that $T \cap (B \times \{\tau\}) = B' \times \{\tau\}$. The number of edges in the cut that are incident to s or t is exactly $n'(|T \cap (A \times \{0\})| + |S \cap (B \times \{\tau\})|) =$

⁸This is the only place in this paper where we will need this overloading of notation.

$n'(|A \setminus A'| + |B \setminus B'|) = n'(k - |A'| - |B'|)$. It implies that the number of edges in G^τ in the (S, T) cut is strictly less than $n'k/2 - n'(k - |A'| - |B'|) = n'(|A'| + |B'| - k/2) \leq h(|A'|, |B'|)$, by the definition of n' . (Note that if $|A'| + |B'| \leq k/2$ then the inequality is trivially true since h is always positive.) Thus, we find a cut in the original graph G^τ of size strictly less than $h(|A'|, |B'|)$ separating $A' \times \{0\}$ and $B' \times \{\tau\}$, a contradiction. This finishes the proof of the lemma. \square

We use the cut-matching game of Khandekar, Rao and Vazirani [KRV09]. In this game, we are given a set V_X of N_X vertices, where N_X is even, and two players: a cut player, whose goal is to construct an expander $X = (V_X, E_X)$ on the set V_X of vertices, and a matching player, whose goal is to delay its construction. The game is played in iterations. We start with the graph $X = (V_X, \emptyset)$.

In each iteration j , the cut player computes a bi-partition (A_j, B_j) of V_X into two equal-sized sets, and the matching player returns some perfect matching M_j between the two sets. The edges of M_j are then added to E_X . Khandekar, Rao and Vazirani have shown that there is a strategy for the cut player, guaranteeing that after $O(\log^2 N_X)$ iterations we obtain a $1/2$ -expander with high probability. Subsequently, Orecchia et al. [OSVV08] have shown the following improved bound:

Theorem 22 (Cut-Matching Game [OSVV08]). *There is a probabilistic algorithm for the cut player, such that, no matter how the matching player plays, after $O(\log^2 N_X)$ iterations, graph X is an $\alpha_{\text{CMG}}(N_X) = \Omega(\log N_X)$ -expander, with constant probability.*

Definition 23. Let E' be a set of edges over K , $\tilde{\tau} > 0$ be an integer. Let $\overline{E'}$ be the set of directed edges obtained from E' by replacing every undirected edge $e = (u, v) \in E'$ with two directed edges (u, v) and (v, u) . An embedding of E' in $G^{\tilde{\tau}}$ is a set $\mathcal{P} = \{P_e : e \in \overline{E'}\}$ of paths, where P_e for a directed edge $e = (u, v)$ is a path connecting $(u, 0)$ to $(v, \tilde{\tau})$ in $G^{\tilde{\tau}}$.

Lemma 24. *There is a randomized algorithm that outputs an $O(\log^2 k)$ -regular $\Omega(\log k)$ -expander $X = (K, E_X)$, and an embedding \mathcal{P} of E_X in G^τ , such that the expected number of paths in \mathcal{P} that use each edge e in G^τ is at most $O(\log^2 k / n')$.*

Proof. We run the cut-matching game over K . Initially, $\mathcal{P} = \emptyset$ and $E_X = \emptyset$.

In the j -th iteration of the game, the cut-player finds a partition (A_j, B_j) of K according to his strategy. Then by Lemma 21, we can find a set \mathcal{Q}_j of $n'|A_j| = n'k/2$ edge-disjoint paths from $A_j \times \{0\}$ to $B_j \times \{\tau\}$ in G^τ , such that every vertex in $A_j \times \{0\}$ is the origin of exactly n' paths and every vertex in $B_j \times \{\tau\}$ is the destination of exactly n' paths. These paths naturally define an n' -regular bipartite graph $H = (A_j \cup B_j, E_H)$ between A_j and B_j , where for each edge $e = (u, v) \in E_H$, $u \in A_j$, $v \in B_j$, e is associated with a unique path $Q_e \in \mathcal{Q}_j$ connecting $(u, 0)$ to (v, τ) in G^τ . We can break E_H into n' matchings between A_j and B_j . Then, the matching player will randomly choose a matching M_j , out of the n' matchings, each with probability $1/n'$. The matching player will play M_j ; so we shall add M_j to E_X .

Let $\mathcal{Q}'_j = \{Q_e : e \in M_j\}$ be the set of paths corresponding to M_j , and let \mathcal{Q}''_j be the set of mirrored paths of paths in \mathcal{Q}'_j . The mirrored edge of an edge $((u, t-1), (v, t))$ in G^τ is the edge $((v, \tau-t), (u, \tau-t+1))$. The mirrored path of a path P is constructed by concatenating the mirrored edges of all edges in P . Thus, if P connects $(u, 0)$ to (v, τ) in G^τ , then the mirrored edge of P connects $(v, 0)$ to (u, τ) in G^τ . Thus, $\mathcal{Q}'_j \cup \mathcal{Q}''_j$ is an embedding of M_j in G^τ . Since paths in \mathcal{Q}_j are edge-disjoint, each edge in G^τ belongs to \mathcal{Q}'_j with probability at most $1/n'$. Thus, each edge belongs to \mathcal{Q}''_j with probability at most $1/n'$. Moreover, $\mathcal{Q}'_j \cup \mathcal{Q}''_j$ causes congestion at most 2 in G^τ . We add $\mathcal{Q}'_j \cup \mathcal{Q}''_j$ to \mathcal{P} .

Considering all the $O(\log^2 k)$ iterations together, \mathcal{P} is an embedding of E_X in G^τ . The paths in \mathcal{P} cause congestion at most $O(\log^2 k)$, and the expected number of paths in \mathcal{P} that use an edge e in G^τ is at

most $O(\log^2 k)/n'$. By Theorem 22, the graph X we obtained is an $O(\log^2 k)$ -regular $\alpha_{\text{CMG}}(k)$ -expander. The algorithm succeeds with constant probability and thus we can repeat the algorithm until it succeeds. The expected number of times we run the algorithm is a constant; this can only increase the expected number of paths in \mathcal{P} that use an edge by a constant factor. \square

We emphasize that we are not interested in the efficiency of the algorithm in Lemma 24 as it is only used for the analysis. Indeed, we need an exponential time algorithm to check whether X is an $\alpha_{\text{CMG}}(k)$ -expander or not since the problem is NP-hard.

We use Lemma 24 to find a d -regular $\Omega(\log k)$ -expander $X = (K, E_X)$, for some $d = O(\log^2 k)$, and an embedding $\mathcal{P} = \{P_e : e \in \overline{E_X}\}$ of E_X in G^τ . Let A be the adjacency matrix of X and λ_2 be the second largest eigenvalue of A . Then, by Cheeger's Inequality, we have $\Phi(X) \leq \sqrt{2d(d - \lambda_2)}$. Thus $\lambda_2 \leq d - \phi^2(X)/(2d) \leq d - \Omega(1)$, since $\phi^2(X)/(2d) = \Omega(\log^2 k)/O(\log^2 k) = \Omega(1)$.

We consider the lazy random walk on X , starting from some vertex $v \in K$. By Theorem 11, the difference between the distribution we obtain after T steps of random walk and the uniform distribution is at most $\sqrt{k} \left(\frac{1+\lambda_2/d}{2} \right)^T$, in terms of the L_1 distance. Notice that $\frac{1+\lambda_2/d}{2} \leq \frac{2-\Omega(1/d)}{2} = 1 - \Omega\left(\frac{1}{\log^2 k}\right)$. If we let $T = O(\log^3 k)$ to be large enough, then the difference is at most $1/(2k)$. Thus, after T steps of the lazy random walk, the probability that we are at each vertex $u \in K$ is at least $1/(2k)$.

Using the random walk, we show how to send $1/(2k)$ units of flow from v to u in X , for every ordered pair $(v, u) \in K^2$. We have k types of commodity, indexed by K . Initially, for every vertex $v \in K$, v has 1 unit of commodity v . At each time step we do the following. For every $v \in K$, and a commodity type $v' \in K$, we send $1/(2d)$ fraction of commodity v' to each of the d neighbors of v ; thus, $1/2$ fraction of the commodity v' will remain at v . After T steps, every vertex $u \in K$ has at least $1/(2k)$ units of commodity v' , for every $v' \in K$. Since X is regular, at each time, the total amount of commodity at each vertex v is 1. In each step, the amount of commodity sent through each edge $e \in E_X$ in each direction is exactly $1/(2d)$.

Now, we can simulate the flow in the time graph $G^{\tau'}$, for $\tau' = T\tau$. Recall that $\mathcal{P} = \{P_e : e \in \overline{E_X}\}$ is the embedding of E_X in G^τ . Initially, for each vertex $v \in K$ and a commodity type $v' \in K$, there is 1 unit of commodity v' at $(v, 0)$. Suppose at the t -th step, we sent x units of commodity v' from $v \in K$ to its neighbor $u \in K$, using edge e in E_X . Let $e' \in \overline{E_X}$ be edge e directed from v to u . Then in graph $G^{\tau'}$, we sent x units of commodity v' from $(v, (t-1)\tau)$ to $(u, t\tau)$, using the path $P_{e'}$, shifted by $(t-1)\tau$ units of time. That is, the shifted path contains $((v', (t-1)\tau + i - 1), (u', (t-1)\tau + i))$, for every edge $((v', i - 1), (u', i))$ in $P_{e'}$. If x units of commodity v' remains at v , then we send x units of commodity v' from $(v, (t-1)\tau)$ to $(v, t\tau)$ using the memory edges at v . Thus, we have a multi-commodity flow in $G^{\tau'}$, where for each ordered pair $(v, u) \in K^2$, we sent at least $1/2k$ units of flow from $(v, 0)$ to (u, τ') .

If an edge $((v', i - 1), (u', i))$ in G^τ is used by p paths in \mathcal{P} , then for every $t \in [T]$, the amount of flow sent through the $((v', (t-1)\tau + i - 1), (u', (t-1)\tau + i))$ is $p/(2d)$. By Lemma 24, the expected amount of flow sent through each edge e in G^τ is at most $1/(2d) \times O(\log^2 k/n') = O(1/n')$, where the expectation is over the randomness of X and \mathcal{P} . Taking all pairs (X, \mathcal{P}) in the probability space (again, we are not interested in the efficiency of the algorithm), and scaling the multi-commodity flow by a factor of $2n'$, we obtain a multi-commodity flow in $G^{\tau'}$, where for each ordered pair $(v, u) \in K^2$, we sent at least n'/k units of flow from $(v, 0)$ to (u, τ') . The flow causes congestion $O(1)$ in $G^{\tau'}$. By, scaling τ' by a constant factor, we can reduce the congestion to 1. This proves that $\tau_{\text{MCF}}(G, K, n') \leq O(T\tau) = O(\log^3 k) \cdot \lceil 4R_c(f, G, K) \rceil = \tilde{O}(R_c(f, G, K))$, finishing the proof of Theorem 19.

5.3 Bounds for ED

The proof of the upper bound in Theorem 2 will crucially use the following result on existence of a small circuit for ED:

Lemma 25. $ED_{K,m}$ has an $(O(km \log k), O(m \log k))$ -bounded circuit.

Proof. We first recall that there exists sorting networks that sort k numbers with $O(k \log k)$ swaps and depth $O(\log k)$ [AKS83]. By swap we mean a gate that takes as input two numbers and outputs the smaller number as the “first” output and the larger number as the “second” output. Note that if the numbers are m -bits then such a swap can be implemented with an $(O(m), O(m))$ bounded circuit. This implies that there exists a $(O(km \log k), O(m \log k))$ -bounded circuit to sort k numbers (each of which is m bits).

Assume that the sorted numbers are $\mathbf{x}_1, \dots, \mathbf{x}_k$. Then note that the final answer is

$$\bigwedge_{i=1}^{k-1} \neg \text{EQ}(\mathbf{x}_i, \mathbf{x}_{i+1}),$$

where $\text{EQ}(\mathbf{x}, \mathbf{y}) = 1$ if and only if $\mathbf{x} = \mathbf{y}$. Note that one can implement the EQ function with a $(O(m), O(\log m))$ -bounded circuit. This implies that we can compute $ED_{K,m}(\mathbf{x}_1, \dots, \mathbf{x}_k)$ with a $(O(km), O(\log km))$ bounded circuit (assuming $\mathbf{x}_1, \dots, \mathbf{x}_k$ are sorted in that order).

Thus, combining the two circuits, we get an $(O(km \log k), O(m \log k))$ -bounded circuit for $ED_{K,m}$, as desired. \square

It is known that $ED_{K,n}$ can be solved by solving $ED_{K, O(\log k)}$ (by using $O(\log k)$ random hashes for each input)– see e.g. [CRR14]. By Lemma 25, there exists a randomized $(O(k \log^2 k), O(\log^2 k))$ -bounded circuit to solve $ED_{K,n}$. Lemma 3 and Claim 8 then show that $R(ED_{K,n}, G, K) \leq \tilde{O}(\tau_{\text{MCF}}(G, K, 1))$. Similarly using Lemma 25 with $m = n$ we have that $R_0(ED_{K,n}, G, K) \leq \tilde{O}(\tau_{\text{MCF}}(G, K, n))$. Note that these upper bounds match the lower bounds in Corollary 20, which in turn proves Theorem 2.

6 Applications

We now consider distributed graph problems. For such problems every player $u \in K$ receives a subgraph H_u and the goal of the players is to compute some (Boolean) function on the overall graph

$$H \stackrel{\text{def}}{=} \bigcup_{u \in K} H_u.$$

We define N_H , M_H and Δ_H to be the number of vertices in H , number of edges in H and the maximum degree in H respectively. We will present our bounds in terms of these parameters (as well as parameters that depend on the underlying topology).

6.1 Distribution of the input

In this section, we tackle issues related to how the inputs $\{H_u\}_{u \in K}$ are represented and distributed among the players in K . We will assume that H_u ’s (and hence H) are presented in the adjacency list representation and that all players know the set of vertices $V(H)$. In other words, the only knowledge that is distributed is the set of edges $E(H)$. There are two natural ways of distributing the edges set that we consider in this section:

1. *Node distribution*: In this case the adjacency list of a vertex is assigned to a terminal in K as a whole. Further, we will assume that for every $u \in V(H)$, all terminals know the location of the assigned terminal for u .⁹ However, only the assigned terminal knows the adjacency list of u .
2. *Edge distribution*: In this case the edge set $E(H)$ is distributed among the k terminals and in this case all the terminals only know about the identity of $V(H)$.

Finally, we will assume that in either distribution all of the H_u 's are roughly of the same size.

Definition 26. A node (edge resp.) distribution of H among the k players is called M -balanced if for every $u \in K$, the size of H_u is at most M .¹⁰

It turns out that one can convert a balanced edge distribution into a random balanced node distribution.

Lemma 27. If H is represented by an $\tilde{O}(M_H/k + \Delta_H)$ -balanced edge distribution then it can be converted into an $\tilde{O}(M_H/k + \Delta)$ -balanced node distribution in $\tilde{O}(\tau_{\text{MCF}}(G, K, M_H/k + \Delta_H))$ rounds of communication. Further, in the latter, every node is assigned uniformly and independently at random to the terminals in K .

Proof. The argument basically follows from a technical result in [KNPR15]. Let $\pi : V(H) \rightarrow K$ be a completely random map (i.e. each vertex is mapped independently and uniformly randomly to K). Then [KNPR15, Lemma 4.1] argues that size of the newly mapped H_u is $\tilde{O}(M_H/k + \Delta_H)$. It is easy to see that we can move from the edge distribution to the random node distribution with a multicommodity flow problem with $\tilde{O}(M_H/k + \Delta_H)$ -bounded demands, which completes the proof. \square

It turns out that the extra pre-processing round complexity of $\tilde{O}(\tau_{\text{MCF}}(G, K, M_H/k + \Delta_H))$ can always be absorbed in the upper bounds that we can prove and so for the rest of the section, when talking about upper bounds we will assume that H is node distributed such that each node is randomly assigned a terminal in K . Note that this implies that our upper bounds hold for worst-case balanced node or edge distribution. However, our upper bounds do not hold when the distribution of H over the terminals is *skewed*. Skew is a known issue in parallel processing and handling it is left as an open problem.

Our lower bounds work for both $\tilde{O}(M_H/k + \Delta_H)$ -balanced node and edge distribution representations. However, unlike the results of [KNPR15], our lower bounds assume a worst-case partition of the input among the terminals.

6.2 Some hard problems

In this section, we define some hard problems that we will reduce to our distributed graph problems.

The two problems, which we dub OR-DISJ $_{K,n}$ and AND-DISJ $_{K,n}$ respectively, informally are the logical OR (and logical AND resp.) of $\binom{k}{2}$ independent copies of the two-party DISJ problem. In particular, each player $u \in K$ gets $k-1$ strings $\{\mathbf{x}_{u,v}\}_{v \in K \setminus \{u\}}$. Then the players want to compute

$$\text{OR-DISJ}_{K,n}(\{\mathbf{x}_{u,v}\}_{u \in K, v \in K \setminus \{u\}}) = \bigvee_{\{u,v\} \in \binom{K}{2}} \left(\bigvee_{i \in [n]} \mathbf{x}_{u,v}[i] \wedge \mathbf{x}_{v,u}[i] \right),$$

⁹This is a relatively mild assumption since these mappings in practical applications are done by publicly known hash mappings.

¹⁰In the case of node distribution, the size of H_u is the sum of the degree of the vertices assigned to u while in the case of edge distribution, the size of H_u is the number of edges assigned to u .

and

$$\text{AND-DISJ}_{K,n}(\{\mathbf{x}_{u,v}\}_{u \in K, v \in K \setminus \{u\}}) = \bigwedge_{\{u,v\} \in \binom{K}{2}} \left(\bigvee_{i \in [n]} \mathbf{x}_{u,v}[i] \wedge \mathbf{x}_{v,u}[i] \right),$$

where for a set S , we use $\binom{S}{2}$ to denote the set of all unordered pairs from S .

We show the hardness of the two above functions by recalling the large communication complexity of two closely related functions in the classical two-party model: Let Alice (Bob) get m strings, $\mathbf{x}_1, \dots, \mathbf{x}_m$ ($\mathbf{y}_1, \dots, \mathbf{y}_m$), with each $\mathbf{x}_i \in \{0, 1\}^n$ ($\mathbf{y}_i \in \{0, 1\}^n$). Let $\text{OR-DISJ-2PARTY}_{m,n}$ denote the problem of determining if any pair of strings $(\mathbf{x}_i, \mathbf{y}_i)$ have a 1 at a common index. Then, the following is a simple implication of Bar-Yossef et.al [BYJKS04].

Theorem 28. $R_{1/3}^{(2)}(\text{OR-DISJ-2PARTY}_{m,n}) \geq \Omega(mn)$.

Similarly, define $\text{AND-DISJ-2PARTY}_{m,n}$ as the 2-party problem of determining if all pairs of strings $(\mathbf{x}_i, \mathbf{y}_i)$ have a 1 at a common index. This is also called the $\text{TRIBES}_{m,n}$ problem. The following establishes its hardness.

Theorem 29 (Jayram et al. [JKS03]). $R_{1/3}^{(2)}(\text{TRIBES}_{m,n}) \geq \Omega(mn)$.

Theorem 19 implies the following results:

Corollary 30. *For any G and K , we have*

$$R(\text{OR-DISJ}_{K,n}, G, K) \geq \tilde{\Omega}(\tau_{\text{MCF}}(G, K, nk)).$$

Proof. Let $f = \text{OR-DISJ}_{K,n}$. Fix some $A, B \subseteq K$ such that $A \cap B = \emptyset$ and $|A|, |B| \leq k/2$. We shall let $\tilde{\mathbf{x}} \in (\{0, 1\}^{nk})^{K \setminus (A \cup B)}$ be an all-0 vector. Note that $f_{A,B,\tilde{\mathbf{x}}}$ is exactly an $\text{OR-DISJ-2PARTY}_{|A| \cdot |B|, n}$ problem. Thus, by Theorem 28, we have that $R_{1/3}^{(2)}(f_{A,B,\tilde{\mathbf{x}}}) \geq \Omega(|A| \cdot |B| \cdot n)$.

Let $\epsilon = 1/3$. Let $n'' = \Omega(n)$ be small enough; let $n' = n''k/2$; let $h(y, z) = n''yz$ for every $y, z \in [k/2]$. Then $\min_{y, z \in [k/2]: y+z > k/2} \frac{h(y, z)}{y+z-k/2} = \min_{y, z \in [k/2]: y+z > k/2} \frac{n''yz}{y+z-k/2} = n''k/2 = n'$, where the second equality holds since $(k/2 - y)(k/2 - z) \geq 0$ implies $yz \geq (y + z - k/2)k/2$, and $y = 1, z = k/2$ implies $\frac{yz}{y+z-k/2} = k/2$.

Thus, if n'' is small enough, then the condition for Theorem 19 holds. Thus, we have

$$\tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R(\text{OR-DISJ}_{K,n}, G, K)).$$

Then $\tau_{\text{MCF}}(G, K, kn) \leq \left\lceil \frac{kn}{n'} \right\rceil \tau_{\text{MCF}}(G, K, n') \leq O(1) \tau_{\text{MCF}}(G, K, n') \leq \tilde{O}(R(\text{OR-DISJ}_{K,n}, G, K))$, by Claim 8. \square

Corollary 31. *For any G and K , we have*

$$R(\text{AND-DISJ}_{K,n}, G, K) \geq \tilde{\Omega}(\tau_{\text{MCF}}(G, K, nk)).$$

Proof. Let $f = \text{AND-DISJ}_{K,n}$. Fix some $A, B \subseteq K$ such that $A \cap B = \emptyset$ and $|A|, |B| \leq k/2$. We shall let $\tilde{\mathbf{x}} \in (\{0, 1\}^{nk})^{K \setminus (A \cup B)}$ be an all-1 vector. Note that $f_{A,B,\tilde{\mathbf{x}}}$ is a $\text{TRIBES}_{|A| \cdot |B|, n}$ problem. Thus, by Theorem 29, we have that $R_{1/3}^{(2)}(f_{A,B,\tilde{\mathbf{x}}}) \geq \Omega(|A| \cdot |B| \cdot n)$. The rest of the proof is the same as that of Corollary 30 and is omitted. \square

6.3 Reductions from OR-DISJ

In this section we consider the following three problems:

Acyclicity. Given H_u to each player $u \in K$, the players have to decide if H is acyclic or not.

Triangle-Detection. Given H_u to each player $u \in K$, the players have to decide if H has a triangle or not.

Bipartiteness. Given H_u to each player $u \in K$, the players have to decide if H is bipartite or not.

The argument below follows from a simple adaptation of the reduction used to prove hardness of these problems for the total communication case in [CRR14].

Theorem 32. *Each of the problems of acyclicity, triangle-detection and bipartiteness for input H on topology G with set of player K needs $\tilde{\Omega}\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H + N_H}{k}\right)\right)$ rounds of communication (even for randomized protocols). Further, these results hold for the case when H is $O(M_H/k + \Delta_H)$ -balanced node (or edge) distributed.*

Proof. We will use a reduction from OR-DISJ $_{K,n}$ to construct an instance H such that either (i) H is a forest or (ii) H has a triangle (depending on the output of the OR-DISJ instance). Note that a protocol for any of acyclicity, triangle-detection or bipartiteness can distinguish between the two cases. Thus, to complete the proof we present the construction of H from a given instance of $\{\mathbf{x}_{u,v}\}_{u \in K, v \in K \setminus \{u\}}$ of OR-DISJ $_{K,n}$. We will argue explicitly for node distribution and mention where the reduction needs to be modified to make it work for edge distribution.

Fix any $u \in K$. We will define the subgraph H_u . H_u is the disjoint union of subgraphs $H_{u,w}$ = $(V_{u,w}, E_{u,w})$ for every $w \in K \setminus \{u\}$. In particular, $V_{u,w}$ consists of one vertex for each domain element of the universe corresponding to the two party DISJ corresponding to (u, w) and two special vertices corresponding to the pair $\{u, w\}$. In other words we have

$$V_{u,w} = \left\{ \bigcup_{i \in [n]} x_i^{\{u,w\}} \right\} \cup \{y^{u,w}, y^{w,u}\}.$$

The edge set $E_{u,w}$ consists of the edge $(y^{u,w}, y^{w,u})$ plus edges between elements that are present in $\mathbf{x}_{u,w}$ and $y^{u,w}$. In other words,

$$E_{u,w} = \left\{ \left(x_i^{\{u,w\}}, y^{u,w} \right) \mid \mathbf{x}_{u,w}[i] = 1 \right\} \cup \{(y^{u,w}, y^{w,u})\}.^{11}$$

See Figure 1 for an illustration of this reduction.

To complete the argument we make the following observations. First if OR-DISJ $_{K,n}(\{\mathbf{x}_{u,v}\}_{u \in K, v \in K \setminus \{u\}}) = 1$, then H has a triangle otherwise H is a forest. Indeed first note that for every $\{u, w\} \in \binom{K}{2}$, the subgraphs $H_{u,w} \cup H_{w,u}$ are node disjoint. Thus, H has a triangle if and only if $H_{u,w} \cup H_{w,u}$ has a triangle for some $\{u, w\} \in \binom{K}{2}$. Next, we note that if $(\mathbf{x}_{u,w}[i] \wedge \mathbf{x}_{w,u}[i]) = 1$ for some $i \in [n]$, then the triple $\{y^{u,w}, y^{w,u}, x_i^{\{u,w\}}\}$ forms a triangle. Otherwise, $y^{u,w}$ and $y^{w,u}$ are connected via edges to disjoint set of the vertices in $\{x_i^{\{u,w\}}\}_{i \in [n]}$, which implies that $H_{u,w} \cup H_{w,u}$ is a forest. This argues the correctness of the reduction.

Second, for every $u \in K$, the player u can construct H_u from its input $\{\mathbf{x}_{u,w}\}_{w \in K \setminus \{u\}}$. Finally, note that in this construction both N_H, M_H are $\Theta(nk^2)$. Further, each H_u is of size $O(nk)$, which is $O(M_H/k + \Delta_H)$. All of the above along with Corollary 30 completes the proof. ¹² \square

¹¹For edge distribution, we assign the edge $(y^{u,w}, y^{w,u})$ to exactly one of H_u or H_w .

¹²The fact that M_H is $\Omega(nk^2)$ follows from the fact that sets in the hard distribution in Corollary 30 have sets whose size is linear in the size of the universe.

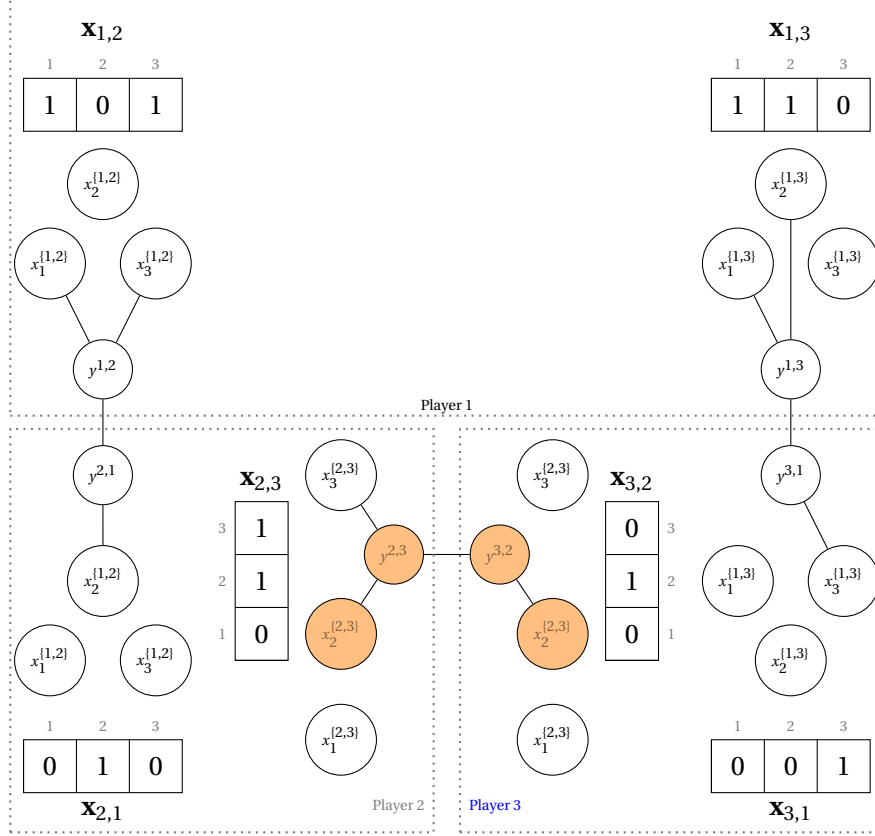


Figure 1: Illustration of the reduction in proof of Theorem 32 for $n = k = 3$. In this example the overall graph H has a triangle and the three participating nodes are colored in orange. (Note that in this case $\text{OR-DISJ}_{\{1,2,3\},3}$ is 1.)

6.3.1 Upper Bounds

We defer the discussion of the upper bounds for acyclicity and bipartiteness to Section 6.4.1.

We next outline a protocol (which is simple generalization of the protocol in [DKO14]) to detect whether H contains a triangle or not.

Proposition 33. *Assuming that for every $\epsilon > 0$, there exists arithmetic circuits of size $O(n^{2+\epsilon})$ for computing $n \times n$ matrix multiplication over \mathbb{F}_2 , the problem of triangle detection on H can be solved with $\tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{(N_H)^{2+\epsilon}}{k}\right)\right)$ rounds of randomized communication.*

Note that the above bound is within any polynomial factor of the lower bound in Theorem 32 for the case of graphs with $M_H \geq \Omega(N_H^2)$.

Proof Sketch of Proposition 33. First, recall that cubing the adjacency matrix of H over the Boolean semiring is enough to detect triangles. This is because a triangle is present if and only if the cubed matrix has a non-zero diagonal entry. It can be shown (see Section 2.1 of [DKO14]) that there exists a randomized reduction of this problem to a few matrix multiplications over the field \mathbb{F}_2 . Now the conjecture about

matrix multiplication yields arithmetic circuits of $O(n^{2+\epsilon})$ size for these matrix multiplications. A further argument shows, exploiting the structure of matrix multiplication [BCS97], that such circuits can be made to have few wires and poly-logarithmic depth. Given such a circuit, an application of our Lemma 3 yields the distributed protocol with $\tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{(N_H)^{2+\epsilon}}{k}\right)\right)$ rounds. \square

6.4 Reductions from AND - DISJ

In this section we consider the following two problems:

Connectivity. Given H_u to each player $u \in K$, the players have to decide if H is connected or not.

Connected Components. Given H_u to each player $u \in K$, the players have to compute the number of connected components of H .

Since a lower bound for connectivity implies a lower bound for the connected components, we only present the lower bound for the latter. This reduction again is a simple adaptation of the corresponding one for total communication in [CRR14].

Theorem 34. *The connectivity problem for input H on topology G with set of player K needs $\tilde{\Omega}\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H + N_H}{k}\right)\right)$ rounds of communication (even for randomized protocols). Further, these results hold for the case when H is $O(M_H/k + \Delta_H)$ -balanced node (or edge) distributed.*

We note that by Proposition 7, the above implies a lower bound of $\tilde{\Omega}((M_H + N_H)/k^2)$ for the case when G is a clique on k terminals. This quantitatively recovers the bound for connectivity proved in [KNPR15].

Proof of Theorem 34. We will use a reduction from AND-DISJ $_{K,n}$ to construct an instance H such that H is connected or not depending on output of the AND-DISJ $_{K,n}$ instance. To complete the proof we present the construction of H from a given instance of $\{\mathbf{x}_{u,v}\}_{u \in K, v \in K \setminus \{u\}}$ of AND-DISJ $_{K,n}$. (The argument holds for both node and edge distributions.)

Fix any $u \in K$. We will define the subgraph H_u . H_u is the union of subgraphs $H_{u,w} = (V_{u,w}, E_{u,w})$ for every $w \in K \setminus \{u\}$. We next present the description of $H_{u,w}$. For the rest of the proof, we will assume that there is a pre-determined total order among the players, i.e. given any two $u, w \in K$, the comparison $u < w$ is well-defined.

In particular, $V_{u,w}$ consists of one vertex for each domain element of the the universe corresponding to the two party DISJ corresponding to (u, w) and two special vertices corresponding to the pair $\{u, w\}$. In other words, we have if $u < w$

$$V_{u,w} = \left\{ \bigcup_{i \in [n]} x_i^{\{u,w\}} \right\} \cup \{\ell^{\{u,w\}}, r\}$$

and otherwise

$$V_{u,w} = \left\{ \bigcup_{i \in [n]} x_i^{\{u,w\}} \right\} \cup \{r\},$$

where the node r is shared across all subgraphs. The edge set $E_{u,w}$ consists of the following: Consider the case $u < w$. If $\mathbf{x}_{u,w}[i] = 1$, then the edge $(x_i^{\{u,w\}}, \ell^{\{u,w\}})$ is present. If $\mathbf{x}_{u,w}[i] = 0$, the edge $(x_i^{\{u,w\}}, r)$ is present. In the other case of $u > w$, edge $(x_i^{\{u,w\}}, r)$ is present if $\mathbf{x}_{u,w}[i] = 1$. See Figure 2 for an illustration of this reduction.

To complete the argument we make the following observations. First note that if AND-DISJ $_{K,n}(\{\mathbf{x}_{u,v}\}_{u \in K, v \in K \setminus \{u\}}) = 1$, then H is connected otherwise H is not. Second, for every $u \in K$, the player u can construct H_u from its

input $\{\mathbf{x}_{u,w}\}_{w \in K \setminus \{u\}}$. Finally, note that in this construction both N_H, M_H are $\Theta(nk^2)$. Further, each H_u is of size $O(nk)$, which is $O(M_H/k + \Delta_H)$. All of the above along with Corollary 31 completes the proof.¹³ \square

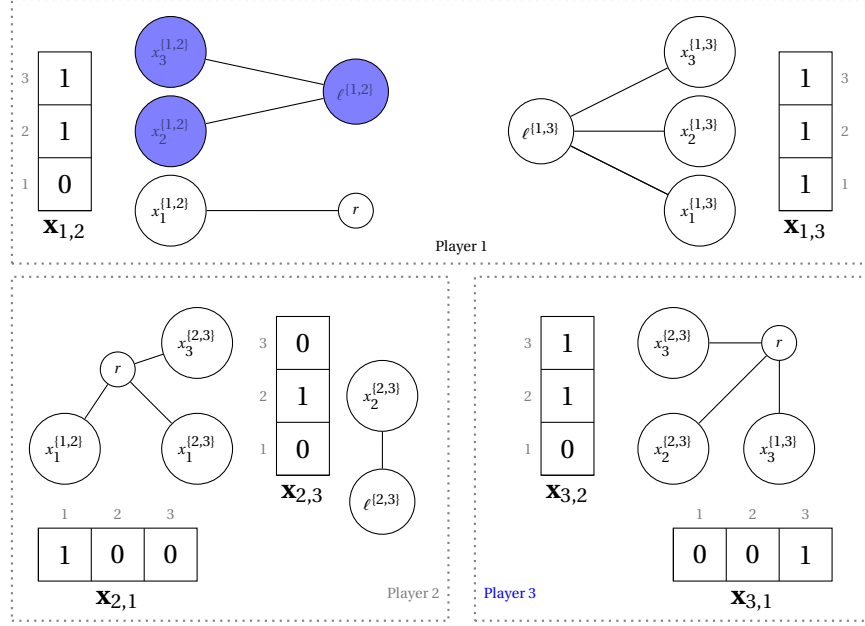


Figure 2: Illustration of the reduction in proof of Theorem 34 for $n = k = 3$. (For clarity singleton nodes in each of the player's subgraphs are not shown.) In this example the overall graph H has two connected components: the nodes shaded blue form one connected component and the rest of the vertices form another connected component. (Note that in this case $\text{AND-DISJ}_{\{1,2,3\},3}$ is 0.)

6.4.1 Upper Bounds

We outline how we can adapt the argument of [KNPR15] to implement BFS in our framework and then argue that for large enough inputs H , the lower bound for connectivity in Theorem 34 is tight. (Recall that we are assuming that the original input H is randomly partitioned across the terminals in a node distribution: we'll call this the random node distribution.)

Theorem 35. *Let H be a random node distributed graph. Then if H is large enough compared to G , we can solve the connectivity problem on H with $\tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H + N_H}{k}\right)\right)$ randomized rounds of communication.*

Before we prove Theorem 35, we will need the following fact about multi commodity flows (which is only needed to prove the tightness of our upper bound):

Lemma 36. *There exists a constant $c > 0$ such that given any G and K , there exists an integer B_0 such that for every $B \geq B_0$ we have that*

$$\tau_{\text{MCF}}(G, K, B) \geq c \cdot \left(\frac{B}{B_0} \cdot \tau_{\text{MCF}}(G, K, B_0)\right).$$

¹³The claim that $M_H \geq \Omega(nk^2)$ follows from the fact that in the hard distribution in Corollary 31, the individual sets are of size $\Omega(n)$.

We remark that the above is not implied by Claim 8. In particular, note that for $n'' \leq n'$, Claim 8 only shows that $\tau_{\text{MCF}}(G, K, n'') \leq \tau_{\text{MCF}}(G, K, n')$, which is not enough to prove Lemma 36.

Proof Sketch of Theorem 35. The basic idea is to run BFS with an arbitrary starting vertex in H . A player $s \in K$ is determined as the start player and s picks an arbitrary node in H_s as the start vertex for the BFS.¹⁴

The idea is to simulate the BFS on H in our framework. Let D denote the diameter of H . We will use the flooding version of BFS. In particular, s sends a *token* to all the neighbors of its chosen vertex in H_s . In future phases, every node in H when it first receives a token, it sends the token to all of its neighbors. If the node has already received the token, then it just ignores the future receipt of the token.¹⁵

Consider the layered graph corresponding to the above run of the BFS on H . For layer $0 \leq i < D$, let H_i denote the subgraph of H that is involved in transfer of token when building layer $(i + 1)$ from layer i . For notational simplicity let n_i be the number of nodes in layer i , $m_i = |E(H_i)|$ and Δ_i denote the maximum degree of any node in layer i . Note since H is randomly node distributed, then so is H_i .¹⁶

Consider the case when we are building layer $(i + 1)$ from layer i . Then the concentration bound proved in [KNPR15, Lemma 4.1] implies that the corresponding multicommodity flow problem is for $\tilde{O}(m_i/k + \Delta_i)$ -bounded demands. This implies that we can simulate the BFS with

$$\sum_{i=0}^D \tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{m_i}{k} + \Delta_i\right)\right) \quad (3)$$

many rounds, where we have that

$$\sum_{i=0}^D m_i = \Theta(M_H), \quad (4)$$

and

$$\sum_{i=0}^D \Delta_i = O(N_H). \quad (5)$$

Now we assume that H is large enough so that

$$\frac{M_H}{D \cdot k} \geq B_0,$$

where B_0 is as defined in Lemma 36. Now note that for every $0 \leq i \leq D$ such that $m_i < \frac{M_H}{D}$, we have that

$$\tau_{\text{MCF}}\left(G, K, \frac{m_i}{k} + \Delta_i\right) \leq \tau_{\text{MCF}}\left(G, K, \frac{M_H}{D \cdot k} + \Delta_i\right).$$

Thus the total contribution of all such i to the bound in (3) is at most

$$\sum_{i=0}^D \tau_{\text{MCF}}\left(G, K, \frac{M_H}{D \cdot k} + \Delta_i\right) \leq O\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H}{k} + N_H\right)\right),$$

¹⁴To be completely correct, we have to make sure that H_s is not empty. But this can be done by a simple leader election algorithm via a Steiner tree style protocol (where each internal node passes on one of the incoming IDs to its parent and the ID picked by the root is declared the leader), which would be smaller than the bound we are after and hence, we should be able to ignore this.

¹⁵The protocol needs to figure out a termination condition. By a simple Steiner tree type protocol one can count the number of nodes that have the token and we can stop if this number does not increase. To prevent overuse of this check, we can perform this in geometrically increasing round numbers.

¹⁶All the bounds used in this proof hold with high enough probability so that we can apply union bound.

where the inequality follows from Lemma 36 and (5). Now for all $0 \leq i \leq D$ such that $m_i \geq \frac{M_H}{D}$, from Claim 8, we have that their contribution to (3) is $O\left(\frac{m_i}{M_H} \cdot \tau_{\text{MCF}}\left(G, K, \frac{M_H}{k} + \Delta_i\right)\right)$. Then by (4) and (5), we have that the total contribution over all such i is also $\tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H}{k} + N_H\right)\right)$.

Thus, we have argued that (3) is upper bounded by $\tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H}{k} + N_H\right)\right)$. If H is large enough, then $M_H/k \geq N_H$, which would imply the claimed upper bound. \square

Next we briefly state how we can use the standard extensions to BFS to extend the protocol in the proof above to work for other problems. To compute the connected components, change the above protocol so that when no more vertices are added to the current component, we check by the Steiner tree based leader election protocol to pick the next starting terminal s and continue till we cannot. For the acyclicity problem, the above protocol should halt whenever a node receives the token more than once. Finally for bipartiteness, we pass two kinds of tokens: one for the odd rounds and one for the even rounds of the protocol and the graph is not bipartite if and only if a node received two different kinds of tokens. (For both the latter two modifications, we might also have to go through all connected components of H .) All this discussion implies that

Theorem 37. *Let H be a random node distributed graph. Then if H is large enough compared to G , we can solve the connected components, acyclicity and bipartiteness problems on H with $\tilde{O}\left(\tau_{\text{MCF}}\left(G, K, \frac{M_H + N_H}{k}\right)\right)$ randomized rounds of communication.*

Acknowledgments

We would like to thank Jaikumar Radhakrishnan for helpful discussions at the early stage of this paper and to Simons institute's Information theory program for providing the venue for these discussions.

References

- [ACLY00] Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. Network information flow. *IEEE Transactions on information theory*, 46(4):1204–1216, 2000.
- [AKS83] M. Ajtai, J. Komlós, and E. Szemerédi. An $O(N \log N)$ sorting network. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 1–9, New York, NY, USA, 1983. ACM.
- [BCS97] P. Burgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*. Springer, 1997.
- [BEO⁺13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 668–677, 2013.
- [BYJKS04] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.
- [CKKV15] Chandra Chekuri, Sudeep Kamath, Sreeram Kannan, and Pramod Viswanath. Delay-constrained unicast and the triangle-cast problem. In *IEEE International Symposium on Information Theory (ISIT)*, pages 804–808, 2015.

- [CM15] Arkadev Chattopadhyay and Sagnik Mukhopadhyay. Tribes is hard in the message passing model. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 224–237, 2015.
- [CR15] Arkadev Chattopadhyay and Atri Rudra. The range of topological effects on communication. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, pages 540–551, 2015.
- [CRR14] Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.
- [DKO14] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 367–376, 2014.
- [DSHK⁺12] A. Das Sarma, S. Holzer, L. Kor, A. Korman, D. Nanongkai, G. Pandurangan, D. Peleg, and R. Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM Journal on Computing*, 41(5):1235–1265, 2012.
- [HKM11] Bernhard Haeupler, MinJi Kim, and Muriel Médard. Optimality of network coding with buffers. In *IEEE Information Theory Workshop (ITW)*, pages 533–537, 2011.
- [HW07] Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007.
- [JKS03] T. S. Jayram, R. Kumar, and D. Sivakumar. Two applications of information complexity. In *STOC*, pages 673–682, 2003.
- [KNPR15] Hartmut Klauck, Danupon Nanongkai, Gopal Pandurangan, and Peter Robinson. Distributed computation of large-scale graph problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 391–410, 2015.
- [KRV09] Rohit Khandekar, Satish Rao, and Umesh V. Vazirani. Graph partitioning using single commodity flows. volume 56, 2009.
- [Lau07] Lap Chi Lau. An approximate max-steiner-tree-packing min-steiner-cut theorem. *Combinatorica*, 27(1):71–90, February 2007.
- [LL04] Zongpeng Li and Baochun Li. Network coding in undirected networks. CISS, 2004.
- [LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
- [LR99] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, November 1999.
- [MRS⁺98] Madhav V. Marathe, R. Ravi, Ravi Sundaram, S.S. Ravi, Daniel J. Rosenkrantz, and Harry B. Hunt. Bicriteria network design problems. *J. Algorithms*, 28(1):142–171, July 1998.

- [New91] Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67 – 71, 1991.
- [OSVV08] Lorenzo Orecchia, Leonard J. Schulman, Umesh V. Vazirani, and Nisheeth K. Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 461–470, New York, NY, USA, 2008. ACM.
- [Pel00] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.
- [PVZ12] Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 486–501, 2012.
- [Raz92] A. Razborov. On the distributional complexity of Disjointness. *Theor.Comput.Sci.*, 106(2):385–390, 1992.
- [RT87] Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.
- [Tiw87] Prasoon Tiwari. Lower bounds on communication complexity in distributed computer networks. *J. ACM*, 34(4):921–938, 1987.
- [WC14] Chih-Chun Wang and Minghua Chen. Sending perishable information: Coding improves delay-constrained throughput even for single unicast. In *IEEE International Symposium on Information Theory*, pages 866–870, 2014.
- [WZ12] D. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *STOC*, pages 941–960, 2012.
- [WZ13] D. Woodruff and Q. Zhang. When distributed computation is communication expensive. In *DISC*, pages 16–30, 2013.
- [WZ14] D. Woodruff and Q. Zhang. An optimal lower bound for distinct elements in the message passing model. In *SODA*, pages 718–733, 2014.
- [Yao79] A. C. C. Yao. Some complexity questions related to distributed computing. In *11th ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

A A useful 2-party communication complexity result

Consider the following 2-party problem. Alice (Bob) gets k (k') strings $\mathbf{x}_1, \dots, \mathbf{x}_k$ ($\mathbf{y}_1, \dots, \mathbf{y}_{k'}$), each n -bit long. They have to determine if one of Alice's strings is the same as that of one of Bob's, i.e. does there exist a pair (i, j) , such that $\mathbf{x}_i = \mathbf{y}_j$. (Note that this is same as checking whether the sets $\{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ and $\{\mathbf{y}_1, \dots, \mathbf{y}_{k'}\}$ are disjoint.) Let us denote this problem as $\text{DISJ}_n^{k,k'}$.

Theorem 38. *The deterministic 2-party communication complexity of $\text{DISJ}_n^{k,k'}$ is $\Omega(\min\{k, k'\} \cdot n)$ for $k, k' = 2^{o(n)}$.*

Proof. WLOG assume $k \leq k'$. Pick some $t = k' - k$ strings from $\{0, 1\}^n$. Let the set of remaining $2^n - t$ strings be called T . Alice and Bob each get k strings from T in the following way: partition T into k equal disjoint chunks, T_1, \dots, T_k . Consider the problem where Alice and Bob each get k strings, $\mathbf{x}_1, \dots, \mathbf{x}_k$ and $\mathbf{y}_1, \dots, \mathbf{y}_k$ respectively, with $\mathbf{x}_i, \mathbf{y}_i \in T_i$. They have to determine if for all i , $\mathbf{x}_i \neq \mathbf{y}_i$. Clearly if Alice and Bob had a deterministic protocol of cost c for solving $\text{DISJ}_n^{k, k'}$, then they would also be able to solve this new problem P with cost c just as a special case. Note that P is essentially $\text{AND} \circ \text{NEQ}$. The i -th NEQ instance has a Boolean matrix of dimension $|T_i| \times |T_i|$ whose rank is $|T_i| = \frac{2^n - t}{k}$. Then, $\text{AND} \circ \text{NEQ}$ matrix is the tensor product of these k matrices. So its rank is $\left(\frac{2^n - t}{k}\right)^k$. Using the fact that communication is lower bounded by the log of rank, we get that $c \geq k(\log(2^n - t) - \log k)$. The claim follows. \square